



**LOW EARTH ORBIT SATELLITE TRACKING TELESCOPE NETWORK:
COLLABORATIVE OPTICAL TRACKING FOR ENHANCED SPACE
SITUATIONAL AWARENESS**

THESIS

Victor A. Salvador, TSgt, USAF

AFIT-ENV-MS-15-M-200

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-15-M-200

LOW EARTH ORBIT SATELLITE TRACKING TELESCOPE NETWORK:
COLLABORATIVE OPTICAL TRACKING FOR ENHANCED SPACE
SITUATIONAL AWARENESS

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Systems Engineering

Victor A. Salvador

Technical Sergeant, USAF

March 2015

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-15-M-200

LOW EARTH ORBIT SATELLITE TRACKING TELESCOPE NETWORK:
COLLABORATIVE OPTICAL TRACKING FOR ENHANCED SPACE
SITUATIONAL AWARENESS

Victor A. Salvador

Technical Sergeant, USAF

Committee Membership:

Dr. Richard G. Cobb
Chair

Dr. David R. Jacques
Member

Lt Col Erin T. Ryan, PhD
Member

Abstract

The Air Force Institute of Technology has spent the last seven years conducting research on orbit identification and object characterization of space objects through the use of commercial-off-the-shelf hardware systems controlled via custom software routines, referred to simply as TeleTrak. Year after year, depending on the research objectives, students have added or modified the system's hardware and software to achieve their individual research objectives. In the last year, due to operating system and software upgrades, TeleTrak became inoperable. Furthermore, due to a lack of student overlap, knowledge of the basic operation of the TeleTrak deteriorated.

This research re-establishes the basic understanding of the TeleTrak System and develops a plan to improve the telescope tracking controller performance. This research uses a subset of the SE process via the operational and system views to understand the tracking subsystem and develop timing tests to observe delays that could impact tracking. Basic tests revalidate and improve understanding of how the Meade telescopes interface with MATLAB. Calibration camera parameters are then refined, allowing a new technique for calibrating existing control algorithms. The analyses of the findings demonstrate that it is possible to improve the tracking controller, but it also uncovers previously undocumented issues with the Meade telescope mount. Future students interested in continuing this research, regardless of which telescope mount is used with TeleTrak, will benefit from the findings of this research.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisors and Matthew Schmunk for their guidance and support throughout the course of this thesis effort.

Victor A. Salvador

Table of Contents

	Page
Abstract	iv
Table of Contents	vi
List of Tables	x
List of Abbreviations	xi
I. Introduction	1
Chapter Overview	1
Background	2
Statement of Problem	3
Research and Investigative Questions	4
Assumptions and Limitations	4
Thesis Overview	5
II. Literature Review	7
Chapter Overview	7
Space Situational Awareness	8
Optical Surveillance Systems	10
AFIT's TeleTrak System	11
Summary	12
III. Methodology	13
Chapter Overview	13
Purpose of the System	13
Current "as is" configuration	15
Analysis of Possible Solutions	17
TeleTrak framework	17

Decomposition.....	18
PC/MATLAB to Meade mount	25
PC/MATLAB to camera.....	29
PC/MATLAB, Meade mount and camera interface.....	34
Summary.....	38
IV. Analysis and Results	39
Chapter Overview.....	39
TeleTrak baseline	39
PC/MATLAB to Meade mount results.....	44
PC/MATLAB to camera results	48
PC/MATLAB, Meade mount and camera interface results	53
Summary.....	60
V. Conclusions and Recommendations.....	62
Conclusions of Research	62
Recommendations	63
Summary.....	63
Appendix A. Multi-TeleTrakNet Concept of Operations	64
Appendix B. TT2k15_serial_tester_v3	75
Appendix C. TT2k15_visible_clock.....	77
Appendix D. How to create a precalcs.mat file	80
Appendix E. How to configure the digital camera	82
Appendix F. A day in the life of iTeleTrak “to-be”	86
Appendix G. Explanation of the sawtooth test	89
Bibliography	92

List of Figures

	Page
Figure 1. SSN Terrestrial Sensor Locations [6].....	9
Figure 2. Sample TLE Set Coordinate System Explanation, Courtesy of NASA [8]	10
Figure 3. High-Level Operational Concept of TeleTrak Net [3]	14
Figure 4. The Main Components of TeleTrak [15].....	16
Figure 5. AFIT TeleTrak Net Systems View (SV-1) revised from [3], Part 1	20
Figure 6. AFIT TeleTrak Net Systems View (SV-1) revised from [3], Part 2	21
Figure 7. "To-Be" TeleTrak Net Activity Diagram (OV-5) [3].....	22
Figure 8. OV-5 of subsystem of research focus.....	23
Figure 9. Sequence diagram of tracking RSO (OV-6c)	25
Figure 10. PC/MATLAB to Meade mount interface delay	26
Figure 11. Flowchart of PC to mount delay test	27
Figure 12. PC/MATLAB to camera interface delay	30
Figure 13. Target traversing at 33 pixels per second.....	31
Figure 14. Flowchart of PC/MATLAB to Camera test.....	32
Figure 15. Desired target size for PC/MATLAB to camera	33
Figure 16. Flowchart of PC/MATLAB, camera and Meade mount test.....	35
Figure 17. Limitation of target view due to NFOV	36
Figure 18. PC/MATLAB to Meade mount response at period = 1.0.....	46
Figure 19. PC/MATLAB to Meade mount response at period = 0.5.....	47
Figure 20. PC/MATLAB to Meade mount response at period = 0.1.....	47
Figure 21. Target displayed on single frame during playback.....	49

Figure 22. Computed delay using a sawtooth curve fit	50
Figure 23. Average and standard deviation of PC/MATLAB to camera delay	52
Figure 24. Camera delay at different frames per second rates.....	52
Figure 25. Static LED target	54
Figure 26. LED on tripod platform	54
Figure 27. Azimuth input generated with the <i>TT2k15_precalcs_statictarget</i> file	55
Figure 28. Sample of azimuth data obtained from single instance (0.8 deg/sec)	56
Figure 29. Azimuth comparisons including best fit.....	57
Figure 30. Azimuth delays	59
Figure 31. Elevation delays.....	59
Figure 32. Azimuth and Elevation delays.....	60

List of Tables

	Page
Table 1. Common movement commands	29
Table 2. Files required for test setup.....	40
Table 3. Required files for GUI operation	41
Table 4. Support files to TeleTrak	41
Table 5. Required files for post analysis.....	42
Table 6. Files not currently in use (requires validation before use).....	43
Table 7. Mean and standard deviation at different periods.....	45
Table 8. Computed mean and standard deviation of the PC/MATLAB to camera delay	51
Table 9. Mean and standard deviation of the Backdelay and Outdelay.....	58

List of Abbreviations

AEOS	Advanced Electro-Optical System
AFIT	Air Force Institute of Technology
AMOS	Air Force Maui Optical and Supercomputing
ASOC	AFIT Satellite Operation Center
Az	Azimuth
CONOPS	Concept of Operations
COTS	Commercial-off-the-shelf
D/T/ID	Detect/Track/Identify
DEC	Declination
El	Elevation
FOV	Field of View
GEO	Geosynchronous Earth Orbit
GEODSS	Ground-Based Electro-Optical Deep Space Surveillance
GUI	Graphical User Interface
INCOSE	International Council on Systems Engineering
JSpOC	Joint Space Operations Center
LAN	Local Area Network
LED	light-emitting diode
LEO	Low Earth Orbit
MATLAB	Matrix Laboratory
MSSS	Maui Space Surveillance System
NFOV	Narrow Field of View

OV	Operational Viewpoint
RA	Right Ascension
RSO	Resident Space Object
SME	Subject Matter Expert
SSA	Space Situational Awareness
SSN	Space Surveillance Network
SST	Space Surveillance Telescope
SV	Systems Viewpoint
TeleTrak	Telescope Tracking
TLE	Two (or Three) Line Element
TW&A	Threat Warning and Assessment
USAF	United States Air Force
USB	Universal Serial Bus

LOW EARTH ORBIT SATELLITE TRACKING TELESCOPE NETWORK: COLLABORATIVE OPTICAL TRACKING FOR ENHANCED SPACE SITUATIONAL AWARENESS

I. Introduction

Chapter Overview

Collaborative optical tracking of low earth orbit satellites has been successfully tested and implemented by previous Air Force Institute of Technology (AFIT) students, relying on the use of MATLAB. While MATLAB allows flexibility from project to project, not all students have robust coding skills. Poor coding practices forced rewrites of previously functioning code due to MATLAB and Windows version updates. The purpose of this thesis is to address and correct changes that rendered the system inoperable, and determine if the tracking subsystem can be improved.

The first task was duplicating the integration of the Telescope Tracking (TeleTrak) system as a single, cohesive system. Second, the feasibility of improving the tracking performance was investigated. As will be discussed, tracking performance has been limited by hardware and software. Future software upgrades may impact tracking performance, and will be discussed in Chapter IV. Lastly, to reduce rework due to poor continuity and MATLAB coding, a framework is established for future AFIT students interested in maintaining the TeleTrak or other computer-controlled telescopes for their own research. This chapter will provide a brief background on the subject, and then explain in greater detail the objective of this thesis, the assumptions made, and limitations.

Background

Advancements in technology have forced our society to look for better and faster ways to communicate not only across the country but across the world in the most expedient way possible. One of the best solutions for accomplishing faster and optimal communication has been through the use of satellites. Today hundreds of satellites as well as debris are orbiting above our atmosphere at Low Earth Orbit (LEO) which is considered between 100 and 1,000 statute miles [1]. Studies have explored how much “real estate” or “blue sky” is available and how close a satellite could be from other objects to include satellites. Satellites are very costly from inception to end of life and normally once a satellite is deployed it is infeasible to send a technical team to fix any issues. Emergent issues are unavoidable and although great care has been placed to deploy an error-free satellite, mishaps happen and debris is created. Debris is also created from launch subsystems and active satellites. According to the Joint Publication 3-14 (Space Operations), “the space environment has unique characteristics that impact military operations.” It also adds that “once considered a sanctuary, space is becoming increasingly congested, contested, and competitive [2].” Because of this, organizations like the United States Air Force (USAF) survey space and monitor satellites and debris in order to avoid collisions.

Since 2007, the Air Force Institute of Technology (AFIT) has been developing research in the field of tracking space objects in LEO using commercial-off-the-shelf (COTS) telescopes and additional components. This simple yet effective system became to be known as TeleTrak which stands for telescope tracking. TeleTrak provides a low cost method to study satellites in LEO and AFIT students have demonstrated the ability

to ascertain some limited information about the object's orbital path, attitude and operational state [3].

Statement of Problem

The inception of TeleTrak was established in 2007 as a low cost tool for students to characterize satellites as well as debris. Between 2012 and 2013, there were no AFIT students conducting research involving TeleTrak, leading to the loss of system expertise. The lack of student subject matter experts within AFIT, combined with required computer hardware and software upgrades, resulted in an inoperable system. In order to make the TeleTrak operational, the research requires re-integration of the MATLAB code with the implementation of a new version of MATLAB (the code worked under the 2009 version, and the current MATLAB version is 2014). Validation is the priority as it is a milestone in order to investigate ways to develop and improve coordinated tracking of space objects using optical telescopes.

Test cases were generated to determine delays between the three main components of the tracking subsystem and then determine if improvement was achieved by comparing to previous subsystem performance metrics. Some challenges of optically tracking space objects with a single telescope have been documented, mainly by demonstrating the effects on the limited characterization and orbit estimation capabilities [4]. To establish a reliable, accurate, and calibrated tracking it is important to develop a method to determine delays within the tracking subsystem.

Research and Investigative Questions

The Air Force Space Surveillance Network (SSN) is the primary entity for keeping track of space objects for our country. However, in 2013 one of their sensor networks known as the Space Fence was deactivated decreasing the information gathered on “unknown” space objects transiting over United States territory. The Joint Publication 3-14, Space Operations defines Space Situational Awareness (SSA) as the “cognizance of the requisite current and predictive knowledge of the space environment and the operational environment upon which space operations depend [2].”

The two primary research goals for this thesis are to:

- Create a baseline framework for future students to enable them to continue using TeleTrak
- Determine and improve current TeleTrak GUI’s method of tracking

Satisfying these primary goals helps enable follow-on research efforts to:

- Supplement/Augment the Air Force Space Surveillance Network
- Establish AFIT’s own database of trackable objects

Assumptions and Limitations

The research conducted in this thesis will be based on a re-use of the operational and systems views created for TeleTrak Network. It is assumed that these SE products accurately represent the systems interface and the operational activities. Furthermore, the research was limited to examining a single integrated telescope and imaging system. Multiple tracking sites cannot be utilized until a single site is working. Additionally, the majority of the testing will be conducted indoors due to generally poor weather in Ohio, which will limit outdoor testing opportunities. It is assumed that the indoor testing is a

suitable surrogate for the outdoor tests. To accomplish the established goals, the following additional assumptions were made:

- MATLAB remains the single software suite used to run TeleTrak; existing code integrates orbit determination, telescope control, video processing / tracking, and recording of both telescope and video data.
- TeleTrak remains focused on using the Meade serial-communication-controlled telescopes. Based on past students' efforts, documented behavior should be treated as suspect and must be re-tested in this project because of software and hardware updates.
- Lab testing is the best place to start, since systematic calibrations haven't occurred in about five years, and some behaviors were never fully determined and/or documented.

The scope of this thesis is limited to examining a single integrated telescope and imaging system. Multiple tracking sites cannot be utilized until a single site is working. The primary limitation in the project is that due to generally poor weather in Ohio, there are limited outdoor testing opportunities. Therefore, in order to accomplish the goals established, the following assumptions were made:

Thesis Overview

This research validates TeleTrak performance previously achieved during past AFIT student research in setting-up the original telescope system for characterization and orbit determination of space objects. After validation, efforts will concentrate on determining if the original single telescope system performance can be improved during tracking. Since this research is concerned with only a single integrated system, remote

operations from the control room and network operations will not be part of this research although the remote operations capability was re-established during this timeframe.

Chapter II covers related research for networking and implementation of telescope systems. Chapter III describes the methodology on how to develop and improve tracking of the system. Chapter IV provides the results and analysis of the findings for the tracking system. Lastly, Chapter V offers insight for future areas of research that may be of interest for individuals exploring similar fields.

II. Literature Review

Chapter Overview

This chapter summarizes previous work to provide the framework for SSA, SSN, TeleTrak and systems engineering. Schmunk [4] noted that in the late 1950s the first satellite could be easily observed flying overhead. Since then, thousands of satellites have been placed in orbit and from these satellites large amounts of debris have been generated. LEO satellites travel at 7000 to 8000 meters per second (which is over 15000 miles per hour), and collisions at these speeds can create additional large clouds of debris. Collision avoidance was the primary focus for the development of SSA. Currently, the SSN tracks more than 22000 [5] objects orbiting the earth using a network of approximately 30 space surveillance sensors [6]. The equipment used for the SSN is mission specific and very costly. AFIT has researched and is exploring a low-cost solution for satellite tracking using optical telescopes called TeleTrak.

To provide background into the research problem, SSA will be addressed first. Narrowing down from SSA, the SSN and its optical surveillance capabilities will be summarized next. Once an understanding of SSA and SSN are complete, the TeleTrak system will be explained. Lastly, a systems engineering approach, which will be utilized to improve and enhance TeleTrak, is briefly described.

Space Situational Awareness

Bennett [7] noted in his research that the SSA doctrine is paramount in the military and can be further divided into four functional capabilities:

- 1) Detect/Track/Identify (D/T/ID)
- 2) Threat Warning and Assessment (TW&A)
- 3) Characterization
- 4) Data Integration and Exploitation

SSA developed out of an initial necessity to maintain an accurate catalog of satellites placed in orbit. However, currently it is known that space is a congested, contested, and competitive environment. It is paramount to track space objects to avoid collisions like the Iridium-Cosmos incident in 2009 which created approximately 1500 pieces of trackable space debris [5].

The USAF is responsible for SSA, and it falls under the directorate of the Joint Space Operating Center (JSpOC) [6]. The subdivision of JSpOC that is directly responsible for SSA is the Space Surveillance Network (SSN). Currently, the SSN is responsible for tracking over 22000 man-made objects, of which approximately 5% are functioning payloads or satellites, 8% are rocket bodies, and the rest are composed of inactive satellites or debris. Figure 1 is the current graphical representation of the SSN, showing the world wide terrestrial location of the sensors. The SSN in this Figure 1 shows three distinctions of which are dedicated (only used for SSA), contributing (provide SSA information, but are not owned or operated by AFSPC) and collateral (provide SSA information and are owned and operated by AFSPC, but have a primary mission other than SSA) [6].

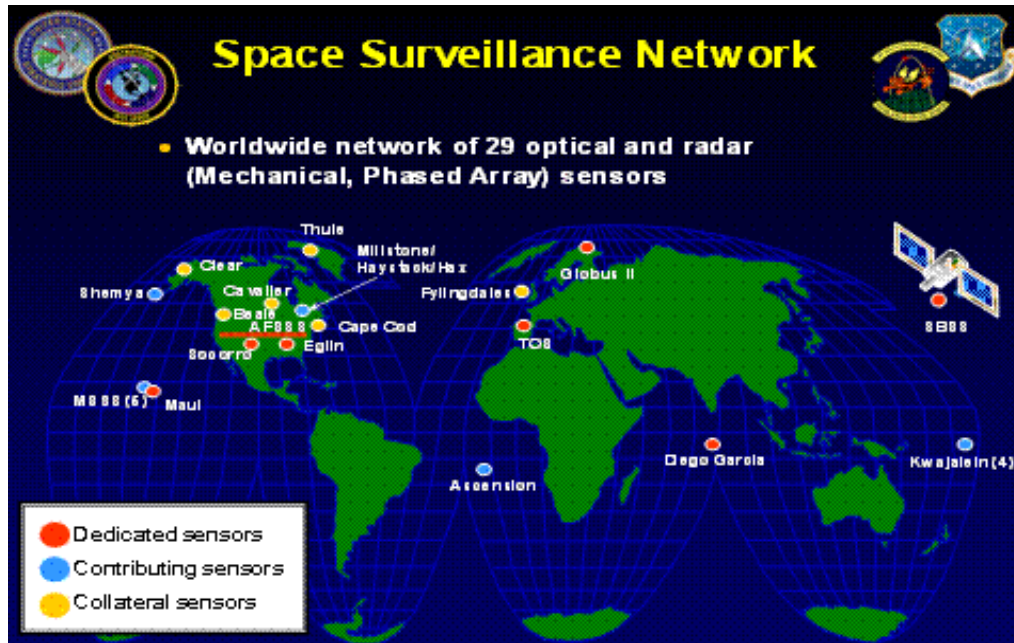


Figure 1. SSN Terrestrial Sensor Locations [6]

Since 2007, AFIT researchers have invested a great amount of time in the SSA field and development of TeleTrak. These efforts are based on the use of telescopes to track satellites while recording images, timing and azimuth and elevation readings. Schmunk was the first AFIT student to investigate a low-cost off-the-shelf telescope system to track satellites in an effort to determine initial orbital estimates. In his research, Schmunk was able to successfully utilize a 10 inch Meade telescope to briefly capture images from a known satellite using the two-line element (TLE) primarily obtained from the Space Track website. A sample TLE is shown in Figure 2. The TLE is used to obtain ephemeris of known orbiting satellites. Other sites that can also provide updated TLE sets are: <http://www.celestrak.com> and <http://www.heavens-above.com>. Using the TLE, a satellite's orbit can be propagated from Epoch to provide an approximate location in the night sky. This information is then transformed into a series

of azimuth and elevation pointing angles as a function of time which TeleTrak then uses for initial acquisition of the satellite image through the telescope's primary and secondary optics.

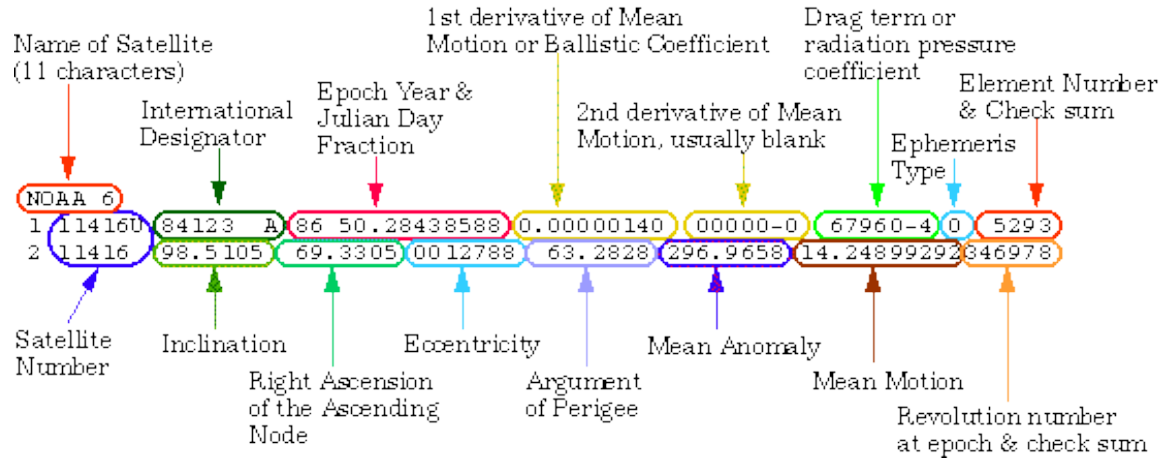


Figure 2. Sample TLE Set Coordinate System Explanation, Courtesy of NASA [8]

Optical Surveillance Systems

In this section a partial description of some of the dedicated SSN sensors will be discussed. The Ground-Based Electro-Optical Deep Space Surveillance (GEODSS) telescopes are able to track small objects, down to the size of a basketball, at more than 20,000 miles away in space. The GEODSS images deep space objects in ranges from 3000 to 22000 miles [9]. These telescopes are specifically designed for deep space and as Briggs [10] noted, these telescopes can operate in rate track as well as sidereal track mode. In rate track mode, the telescope follows the satellite. In sidereal track mode, the telescope stays fixed on a star and the satellite appears as a streak in the image.

The Maui Space Surveillance System (MSSS) utilizes several optical systems to include the Advanced Electro-Optical System (AEOS), a 1.6 and 1.2 meter telescope, and

the RAVEN telescope system [7, 10]. According to Briggs, the RAVEN telescope is “one of the most accurate metric sensors in the SSN” [10]. This implies that the RAVEN is capable of obtaining high accuracy angular observations with a standard deviation of approximately one arc per second. This capability is used to acquire viewing angles for improved satellite orbit determination accuracy [11]. It is important to note that although these sensors work well, they are task saturated and additional sensors are needed.

AFIT’s TeleTrak System

In 2007 Schmunk was able to perform a proof of concept of a low-cost telescope system that utilizes off-the-shelf components. He also introduced the idea of tracking satellites in Low Earth Orbit (LEO) with a MATLAB based program. The optical tracking of space objects using COTS telescopes at AFIT is known as TeleTrak. Several students followed Schmunk’s initial system and added or used some of the capabilities of the system. For instance, Carlton [10] attempted to coherently image LEO satellites using a technique called “lucky imaging” which meant that a satellite could be better imaged and characterized by stacking images of the same satellite. This method allowed the system to observe attitude, payload mission and identification. Briggs implemented a better orbit estimation using an Angles-Only Orbit Determination, and Gresham [7] added a closed-loop controller on the image that would allow higher fidelity of the captured images. Next, an improved closed-loop system was introduced by Graff [8] as well as a remote operation concept utilizing a Meade LX200GPS mount that could be remote controlled. In 2011 Driskell [9] suggested that TeleTrak could be networked and in 2012 Schreiner [1] created a discrete event simulation model to demonstrate that in

theory a TeleTrak Network (TeleTrak Net) could be used to track up to 50 satellites per night with the use of up to 3 telescope systems at multiple locations.

Summary

This chapter addressed previous work related to SSA. Under the direction of the Joint Space Operating Center (JSpOC), SSN has been given the responsibility to track and monitor over 22000 space objects. TeleTrak promises to provide a low-cost off-the-shelf system to characterize space objects. Since the inception of TeleTrak, several students have added features and the continuity which was provided via student to student ended in 2012 when a break in the use of the system occurred. To help alleviate this, a combination of operational and system views will be applied to set up an investigative track testing. The next chapter will cover the proposed method to improve the TeleTrak system.

III. Methodology

Chapter Overview

In this chapter a systems engineering approach will be implemented to establish the purpose of the system, the current system configuration labeled “as is” and the desired system labeled “to be.” With this information, a distinction of the two configurations can be observed and used to establish a starting point. Possible solutions that will meet the requirements for the new system will be analyzed. The purpose of the system will be discussed using a concept of operations provided by the Subject Matter Experts (SMEs). The current system configuration will be used to determine the operational status, and with this information a course of action can be determined by identifying similarities and gaps within the proposed system.

Purpose of the System

As technology in space advances, the need to monitor space assets as well as the space debris created increases. AFIT students in the Aeronautical and Astronautical Engineering department have developed an optical telescope system concept that has proven to be valuable in the characterization of space objects. As discussed in the previous chapters, this optical system came to be known as TeleTrak.

A high-level operational concept of the “to-be” system will be required to understand the decision making process in the “as is” system. This high-level operational concept is depicted in Figure 3 where the AFIT Space Operations Center (ASOC) is the center of operations and where the decision making process takes place. The telescope is located on the rooftop of AFIT’s building 640 which can be commanded

from the ASOC. The collection of data takes place at the telescope location and is transferred to the ASOC where the data are analyzed and processed. These data are added to the AFIT database to create a baseline for reference.

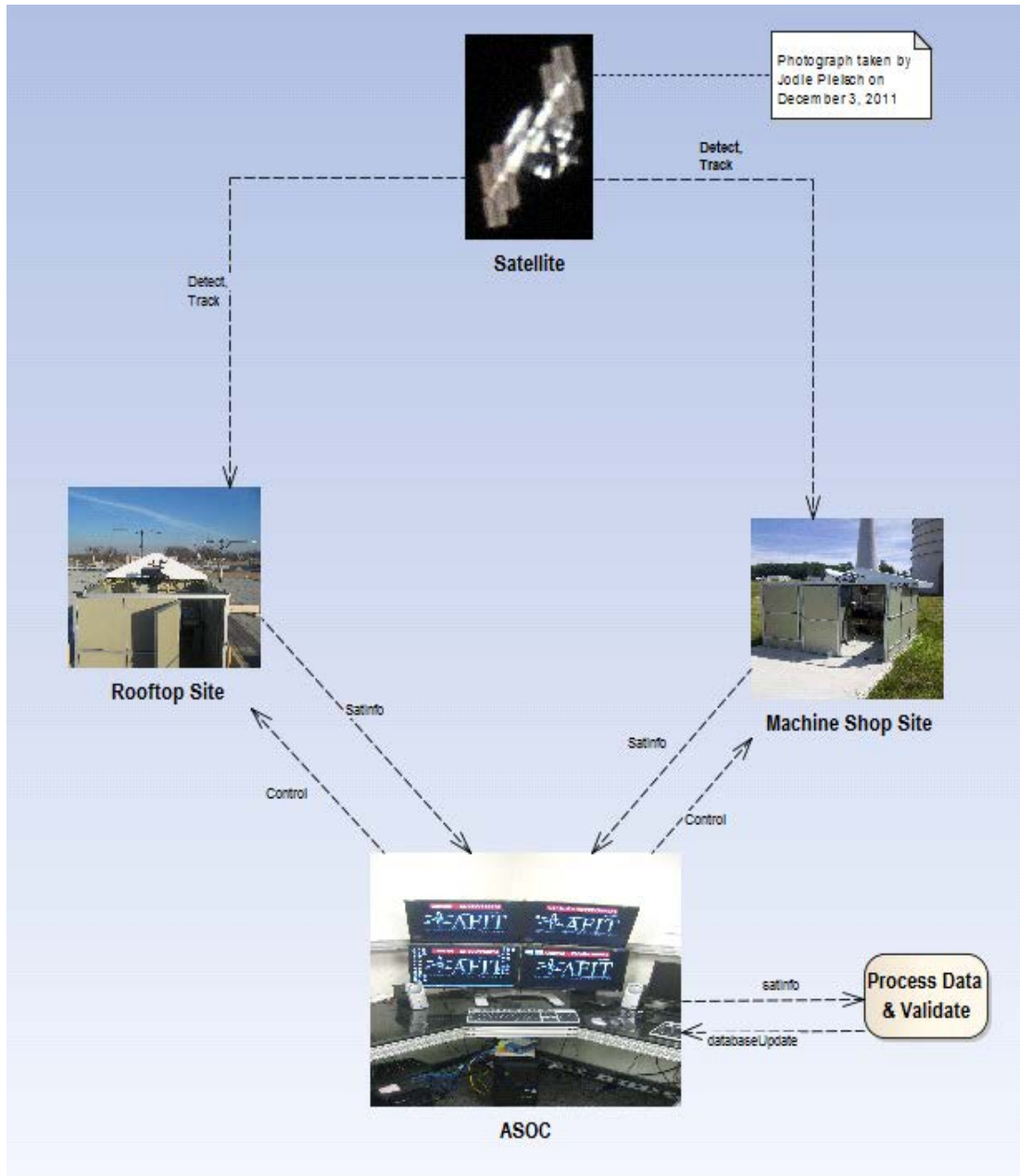


Figure 3. High-Level Operational Concept of TeleTrak Net [3]

Current “as is” configuration

The TeleTrak concept came to fruition as a need to develop a low cost system using only commercial-off-the-shelf equipment. The components depicted in Figure 4 are required by TeleTrak to track space objects. The computer/laptop running MATLAB is used to integrate the telescope and camera. The camera is required to take images of the selected object through the telescope and send the images to the computer. The 80mm scope is currently the primary tracking optic used because it provides a wider field of view than the main optic; this scope is also used to calibrate the telescope. The telescope’s main optic is used for observations. It can be connected to a camera that is either integrated as a tracking optic or recorded separately. The telescope uses a mount that performs azimuth and elevation maneuvers. The rate at which the telescope moves depends heavily on the orbit regime of the tracked object – faster speeds for LEO objects, and near static settings for GEO objects. This configuration shows the main components used. However, it is important to mention that the computer and laptop are the two components that have been upgraded which imply a new operating system, and a new version of MATLAB as well as compatible drivers for the Meade mount and the camera. The upgrade of the computers is key to the overall functionality of TeleTrak.



Figure 4. The Main Components of TeleTrak [15]

Below are the hardware components used for this research, which are the minimum to operate TeleTrak.

- Tracking optic Telescope (Orion 80mm for this research)
- Telescopes mount Meade XL200GPS used for maneuvering
- PC with MATLAB Windows based OS is used to run MATLAB
- USB digital camera Philips SPC 900NC PC camera
- USB cable Used to interface between PC and digital camera
- USB to serial cable Used to interface between the telescope mount and PC as the Meade only uses serial communications
- Power supply for the components

Preliminary assessment of TeleTrak has shown limited functionality and the followings are shortcomings of the system.

- The MATLAB GUI has lost functionality;
- Inability to network two or more telescope systems.

Analysis of Possible Solutions

Although the future “to be” system is the ideal, this research will only focus on developing a baseline to re-establish the TeleTrak GUI (graphical user interface) functionality. *TT2k15_trackgui* is not the main file but is the one the user runs, and it generates the user interface. This will be done by determining the minimum required hardware and software to successfully operate TeleTrak. Once the baseline has been established, the main focus will be on attempting to improve the tracking capability of TeleTrak created by Schmunk and Briggs. Although the system was thought to have an operational tracking method, it is the intent of this research to verify if it can be improved by systematically developing a method that determines the total mean delay of the system, which has a great impact on how the algorithm performs calculations to predict the future position of space objects. Miscalculating the “future” position of the tracked object can make the telescope “jerky.” This means that if the system overshoots the desired position of the telescope, and then it determines that it went too far, the system will re-calculate and move “back,” thus causing a ripple effect. Image capturing during such motion becomes a challenge as well as the analysis of the data.

TeleTrak framework

Besides the upgrade of the operating system and MATLAB it was necessary to understand if there were any additional changes to the system. In order to determine the requirements for TeleTrak, reference documents from previous AFIT research were examined. A TeleTrak CONOPs was developed by Schreiner based on the CONOPs of the Space Surveillance Telescope (SST) [3]. The TeleTrak CONOPs is located in

Appendix A. With the CONOPs, an OV-1 is developed to show the multi-telescope tracking network system. The OV-1 shows detecting, tracking, identifying and cataloging space objects. Also, derived from the OV-1, a Use-Case was depicted by Schreiner[3] which includes all the actors and capabilities involved for a single system. With this information, the decomposition of the system is performed to examine each subsystem and determine the proper course of action to establish and maintain continuity for the lifecycle of TeleTrak.

Decomposition

In order to develop an understanding of any subsystem, it is important to be able to see the “big-picture” and then slowly move to lower system levels until the desired operational level is accomplished. The decomposition of the system starts with the high-level operational concept (OV-1). From the OV-1 the systems view (SV-1) and the operational view (OV-5b) may be derived. This SV-1 will help determine the nodes of the systems, and the interfaces between the system nodes. Next, the operational view (OV-5b) activity diagram of the “to-be” TeleTrak shows the relationships among activities, inputs and outputs. Within the OV-5b, we can further concentrate on a set of specific activities that provide input/output relationships. The result of this is an operational view (OV-6c), which is a sequence diagram designed to trace the sequence of events. Efforts for this research will use the OV-6c to:

- Improve the current tracking system (reliability, smoothness, etc.). The results from tracking testing are compared to earlier results and conclusions observed in three subsystems:
 - Mount
 - Digital Camera
 - GUI (especially ease-of-use or removing irritating behavior).

- Improve test scripts and recorded data to make it easier to collect and compare results from more than one telescope, or from different types of telescopes. In this project, iterations of the “playback” code accomplish this goal.

As stated earlier, the main components are a digital camera, Orion telescope, Meade telescope and mount, and MATLAB code on a PC. It has been addressed by Schmunk (in 2007) that a delay in the system would create errors in the calculation, but he thought that this delay would be difficult to measure. These errors could make the telescope overshoot, resulting in a system that constantly points either ahead or behind the target. With new equipment and better understanding of the delays, an improved controller can be developed.

With the aid of the high-level operational concept, several views can be derived. The systems view (SV-1) depicted in Figure 5, is extremely useful because it aids in the identification of systems, items and interconnections. This “to-be” SV-1 (part 1 and 2) was updated from Schreiner’s [3] as some hardware had changed or was no longer needed. Shreiner had additional hardware components in the SV-1 like USB cables which can be noted as connecting links. The activity diagram (OV-5b) provides a more detailed view of the relationship among inputs and outputs. Note: this OV-5b shows that a bad weather forecast means that there will be no opportunity to observe the desired space object and that the operator will have to try to observe again next night.

For this research the area of interest is inside the red box in Figure 7. This area is responsible for searching, detecting and tracking a space object and is shown in Figure 7 and Figure 8.

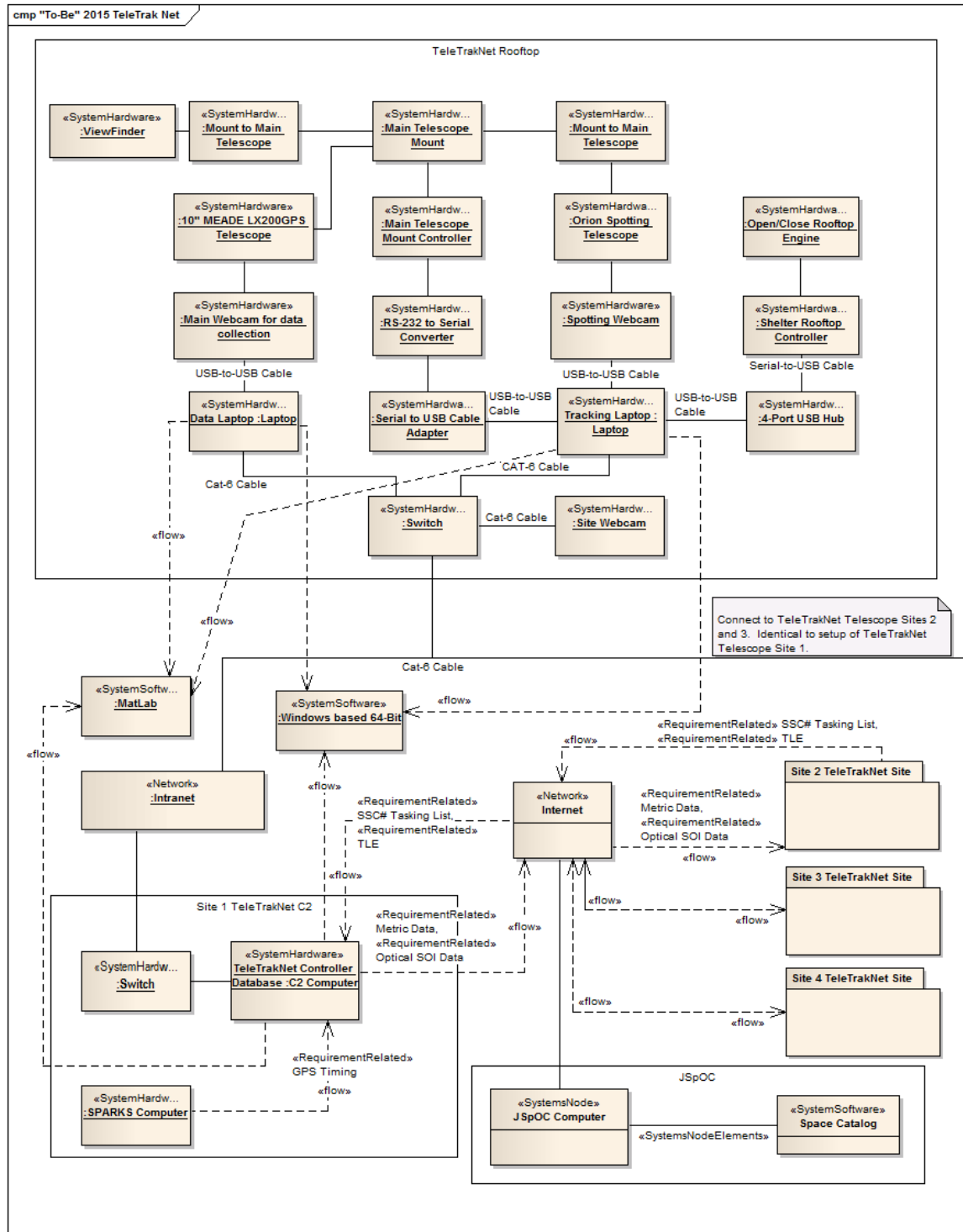


Figure 5. AFIT TeleTrak Net Systems View (SV-1) revised from [3], Part 1

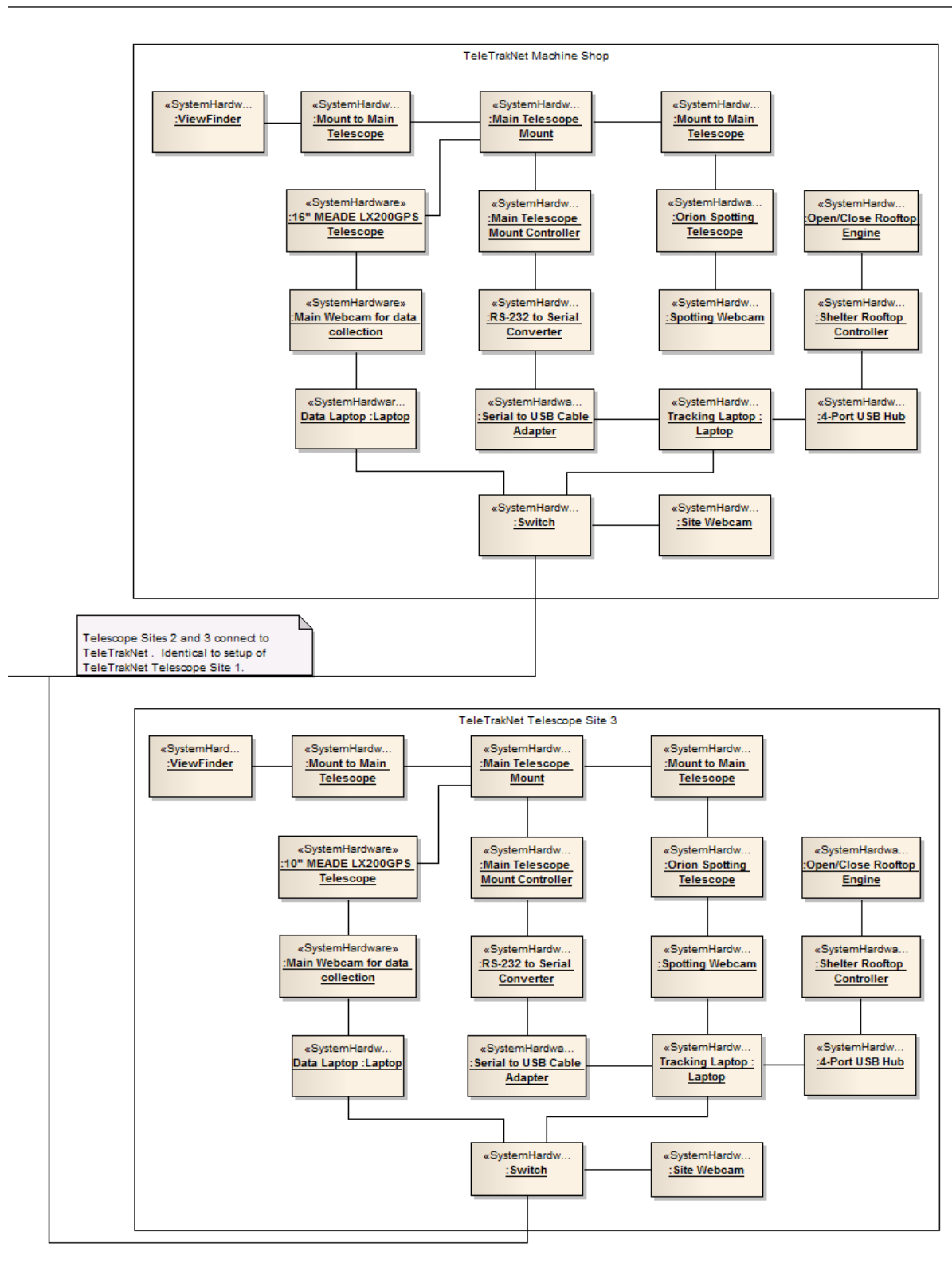


Figure 6. AFIT TeleTrak Net Systems View (SV-1) revised from [3], Part 2

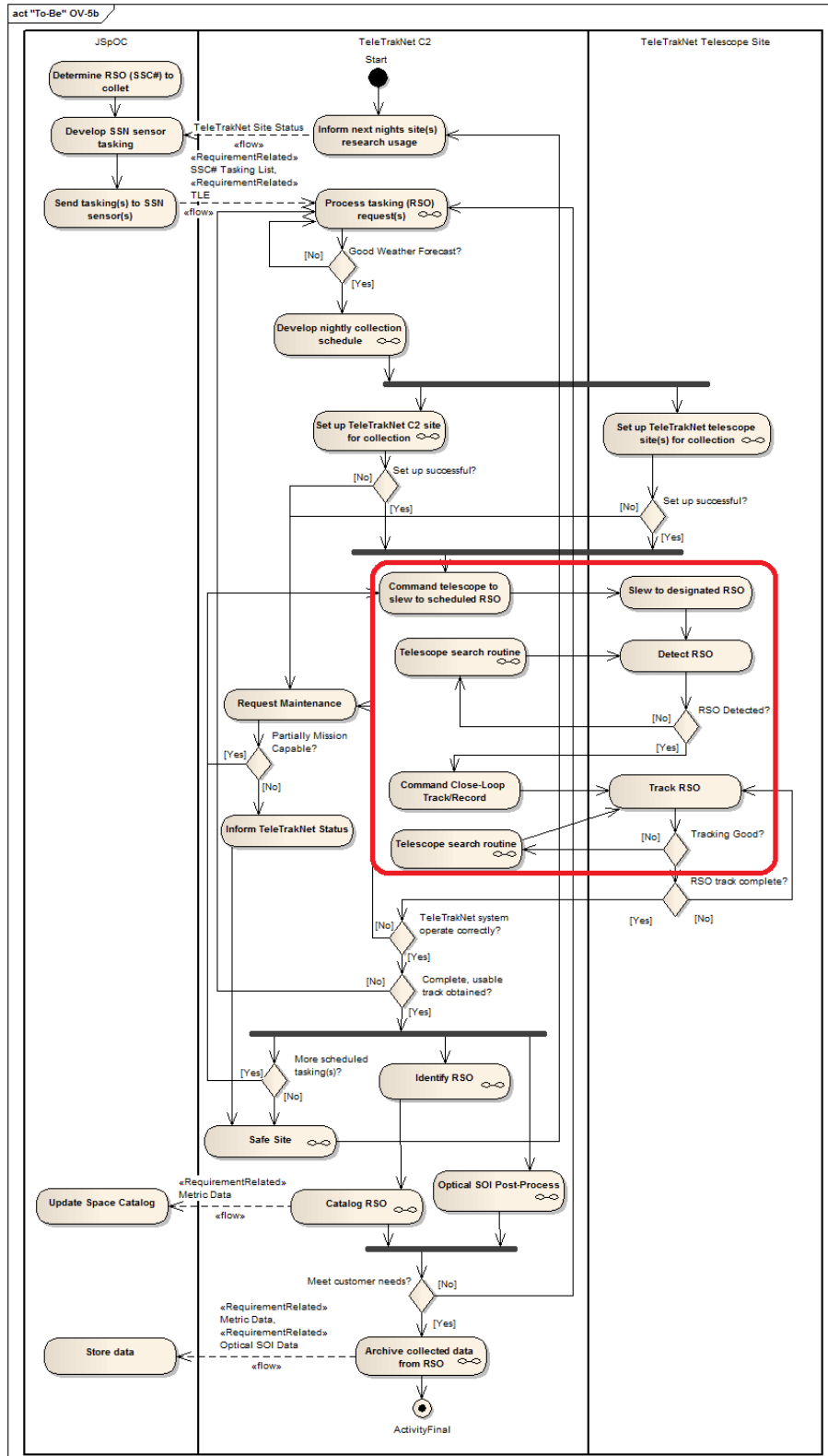


Figure 7. "To-Be" TeleTrak Net Activity Diagram (OV-5) [3]

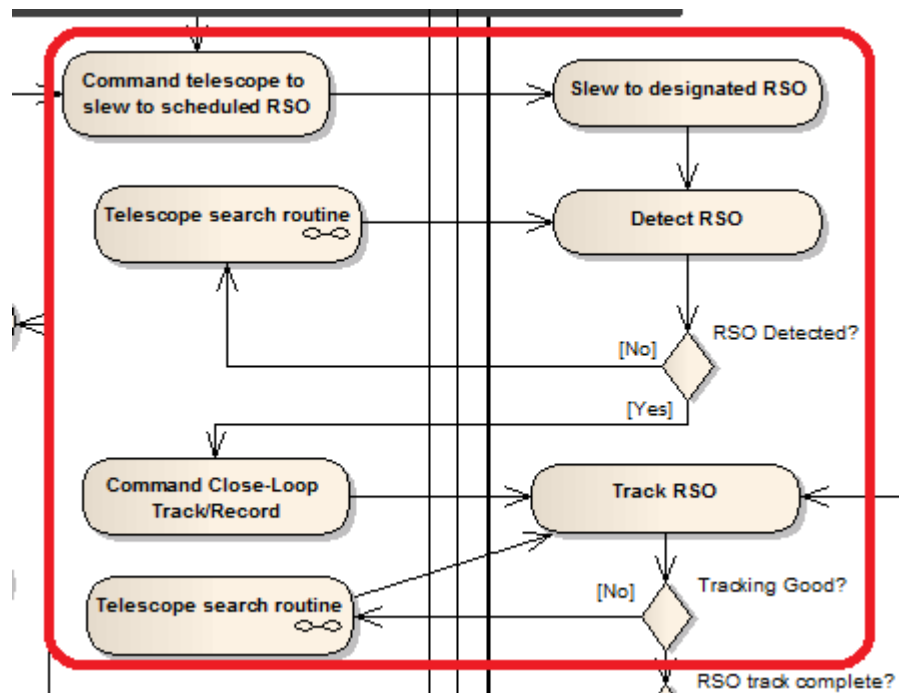


Figure 8. OV-5 of subsystem of research focus

The sequence diagram, OV-6c in Figure 9, specifically shows the area of interest to develop an improved tracking controller by minimizing delays in the tracking system. The OV-6c will be further decomposed to determine what sections induce the largest delay and also the causes for each delay. .

The sequence diagram shows the following flow of data:

- The PC/MATLAB initiates a request to the Meade mount to find out the current pointing position
- The Meade mount then responds with an angle position of where it “thinks” is pointing to in the format DDD MM SS
- The PC/MATLAB establishes communication with the camera
- The camera sends images of what it “sees” to the PC/MATLAB

- The PC/MATLAB compares the images received and determines the new position (if any) of the RSO
- The PC/MATLAB sends a slew command to the Meade mount to point to the previously computed position

Knowing the flow of information is useful because this research is interested in finding the delays in the tracking subsystem. This sequence diagram will be divided into three separate sections: PC/MATLAB to Meade mount, PC/MATLAB to camera and the PC/MATLAB, Meade mount and camera combined as shown in Figure 9

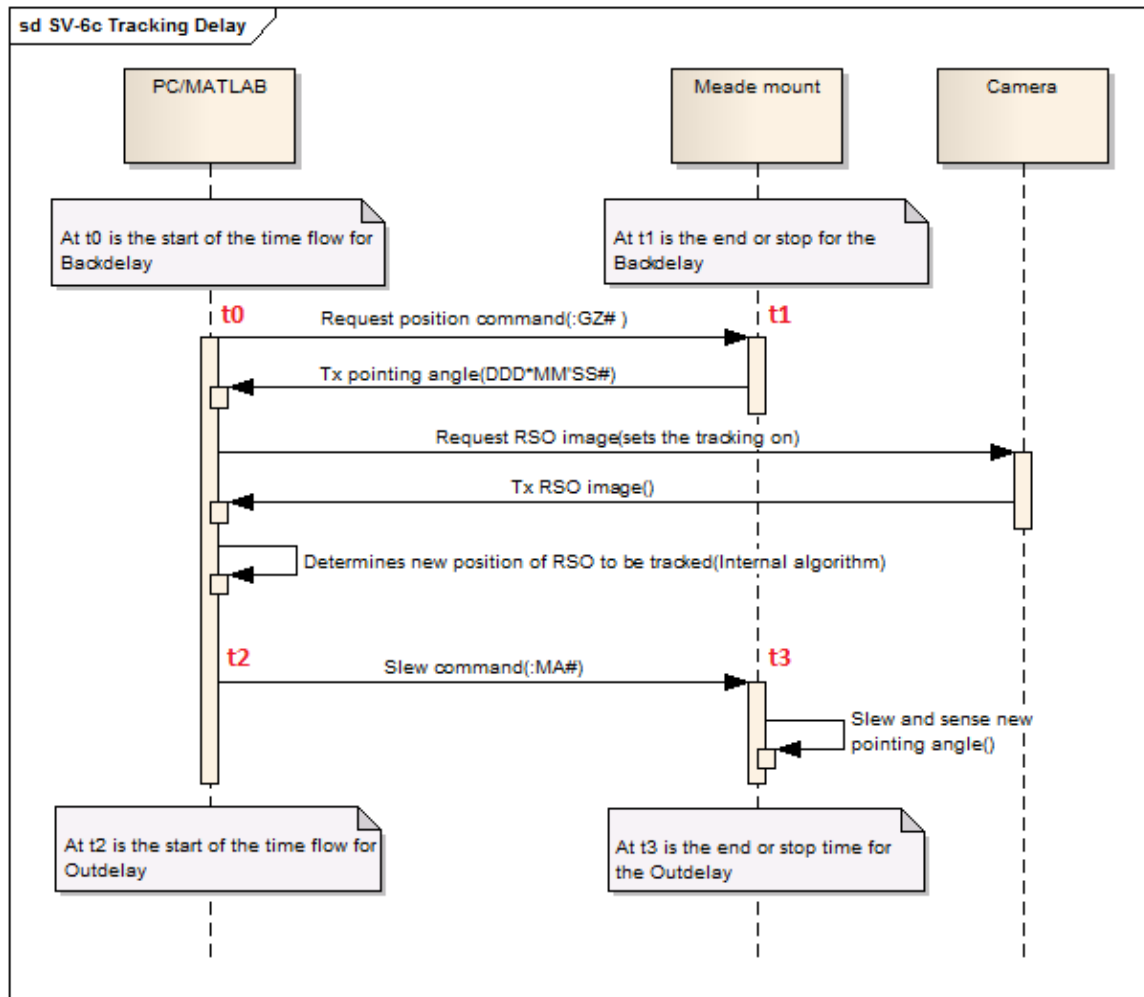


Figure 9. Sequence diagram of tracking RSO (OV-6c)

PC/MATLAB to Meade mount

The objective is to find adequate, but not overwhelming communication rates between the PC/MATLAB and the Meade mount. Requesting or sending a slew command to the Meade mount every second may be simple for the mount to achieve; however, it is impractical because space objects may move through the optic's field of view relatively fast and the telescope may never catch the object. On the other hand, requesting or sending position and slew commands too fast may overwhelm the Meade

mount and the mount may report erroneously or drop requests altogether. This mount uses serial asynchronous communication (a necessity for the PC/MATLAB interface) which can present challenges as it lacks a handshake¹ capability.

It is also thought that the Meade's mount has limitations but it is unclear what they are. A simple, yet effective test may provide the necessary information for tasking the Meade's mount more effectively. One possible system limitation is a long delay between the PC/MATLAB and Meade mount communication resulting in poor tracking ability. Figure 10 shows the flow of information between the PC/MATLAB and the Meade mount in a straight line to interpret the total time it takes for a single command and determine the possible causes for this delay.

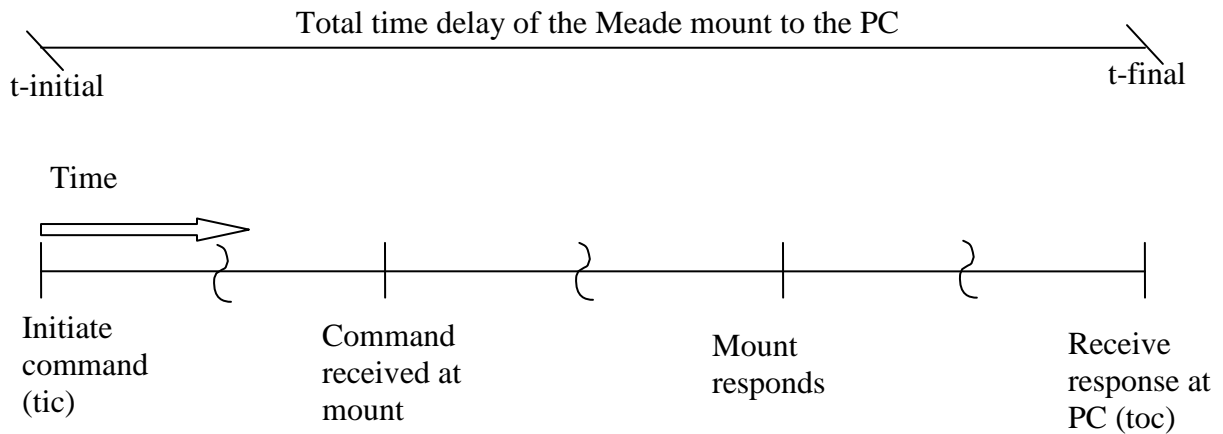


Figure 10. PC/MATLAB to Meade mount interface delay

In order to create a MATLAB script to provide the desired communication between these two components, a flow chart has been created to identify the steps and

¹ In data communications, a handshake is a process to initiate and control communication flow however the Meade mount doesn't handle handshaking and instead when communication is established, MATLAB sends commands and expects the Meade mount to perform them correctly.

procedures that will be taken to perform the test. First, communication with the telescope has to be initiated. Second, MATLAB establishes the mount settings. The procedure will change how often (what period) a command is sent to the mount and determine how often is sufficient to obtain accurate response from the mount. This scenario will be performed around 40 times each period.

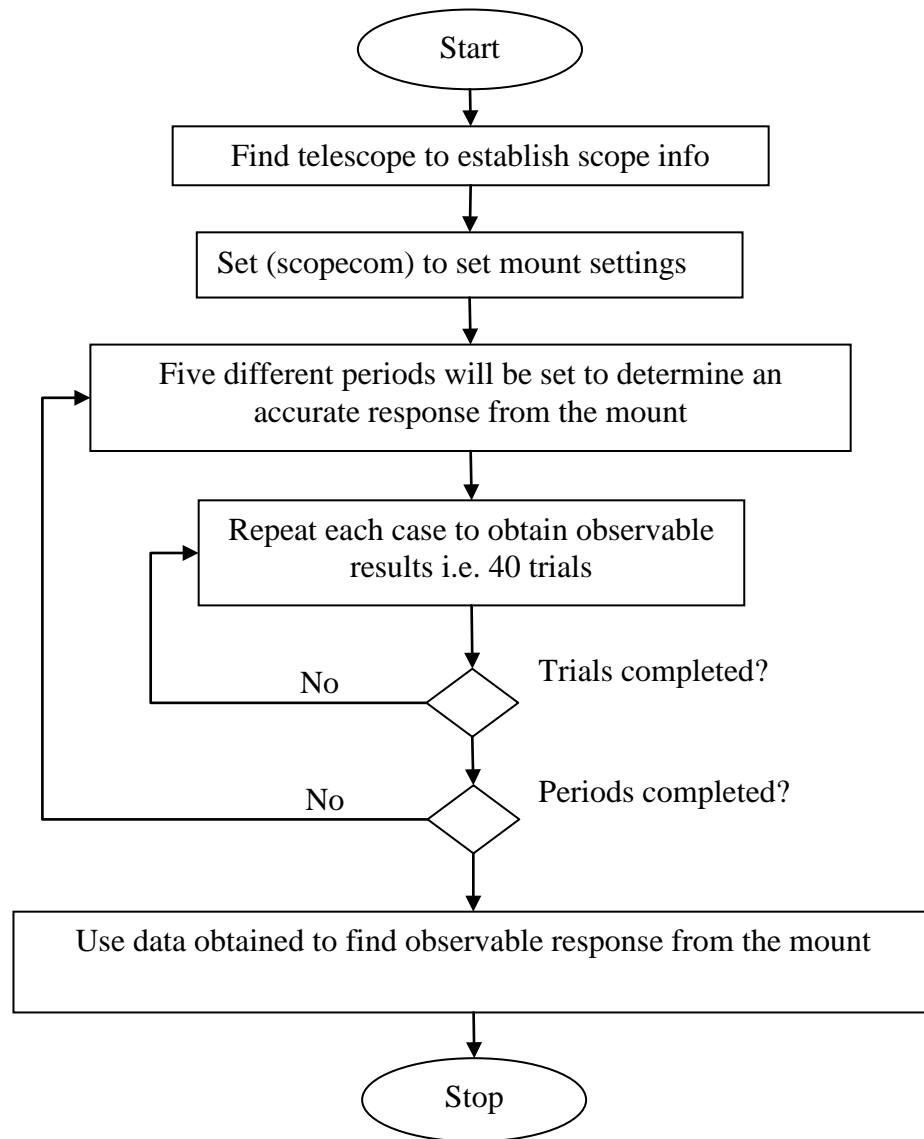


Figure 11. Flowchart of PC to mount delay test

Within the test script, a “tic” is added to start a clock timer of when the command is sent and a “toc” is used to stop the timer for when the telescope replies to where it thinks it is pointing. Determining the overall time is the objective of this test. At lower loads or requests it is expected that the mount will report each request without any conflicts; however, at higher loads it is expected the mount will start dropping requests and/or providing erroneous positions. Note that Richard Seymour, a hobbyist for this particular telescope has mentioned that “the LX200GPS firmware can “safely” handle about four commands per second (i.e. two “what RA are you at? what DEC are you at?” pairs, for example)” [12]. If this is correct then it should be safe to send two commands per second to the Meade mount.

This script uses the same movement techniques previously established for TeleTrak, with all options listed here for completeness in Table 1. Simple read commands from the telescope can be set using the *fscanf(scopecom,"%c",10)*, which reads 10 bytes and

fprintf(scopecom,[':RA',sprintf('%0.2f',azrate_send),'#:M',slewdirection_az,'#:GZ#'])

to send commands to the Meade mount. Note: the “scopecom” is the MATLAB handle assigned to the serial port associated with the mount.

Table 1. Common movement commands

The 'M' commands are for movement		'Q' commands are to halt movement	
':RADD.D#:Me#'	Sets az travel rate; second command causes scope to move clockwise at that rate. Setting rate by itself does not cause rate change.	':Q#'	Halts all movement, used when “quitting” an operation. Setting a zero rate using 'M' commands also halts movement.
':RADD.D#:Mw#'	Moves az axis ccw	':Qe#'	Halts east movement
':REDD.D#:Mn#'	Moves el axis up	':Qw#'	Halts west movement
':REDD.D#:Ms#'	Moves el axis down	':Qn#'	Halts north movement
		':Qs#'	Halts south movement

With the basic knowledge of how to access the mount, the script will be written to test the frequency at which the PC/MATLAB should communicate with the mount. This script (*serial_tester_v3*) can be found in Appendix B. Data obtained from the test were analyzed and presented as a response plot.

PC/MATLAB to camera

To determine the interface delay between the camera and the GUI, it is necessary to accomplish steps similar to the previous test. The camera may be the most important factor in this process as several different cameras are planned for use in the near future. However, if the method for this camera proves successful then it can be used to calibrate other cameras. In order to determine the time it takes from the camera to the GUI, this research develops a method to create a time-dependent target with the same MATLAB instance² ensuring a direct clock correlation between the target and the recorded video.

² Instance in this context refers to simultaneously run the *TT2k15_visible_clock* with the GUI to ensure same time stamp.

Figure 12 shows the total time that takes for an image to be captured and processed by the PC/MATLAB. This time is captured in a time stamp.

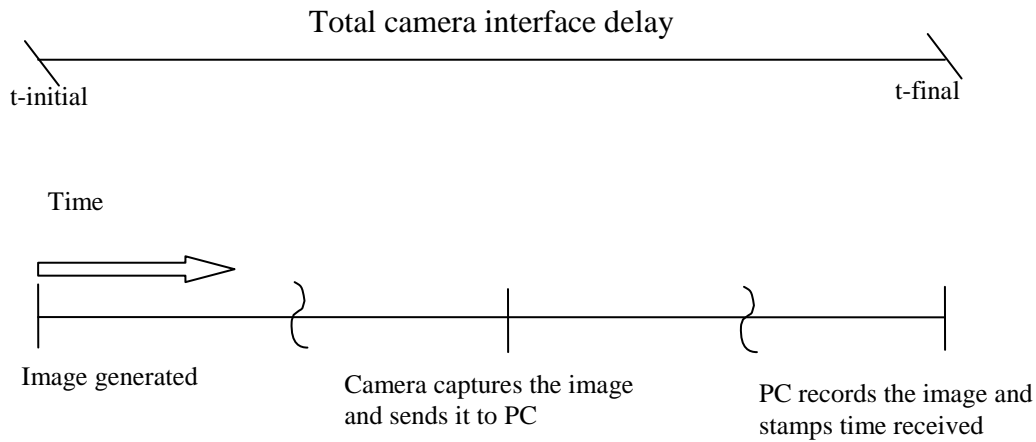


Figure 12. PC/MATLAB to camera interface delay

The scenario will start with the *TT2k15_trackgui* and the camera already connected. The camera will be pointing at a second monitor attached to the same PC that is running the *TT2k15_trackgui*. The second monitor will display a target (a bright pixel) that moves from left to right and travels almost the length of the FOV, but not more than the FOV. The target will be set to move from left to right taking exactly one second per cycle as depicted in Figure 13. By maintaining the same clock on both the target and the GUI, it is expected to be able to obtain the time it takes for the image to make it to the GUI, and consequently the delay between the camera and the GUI.

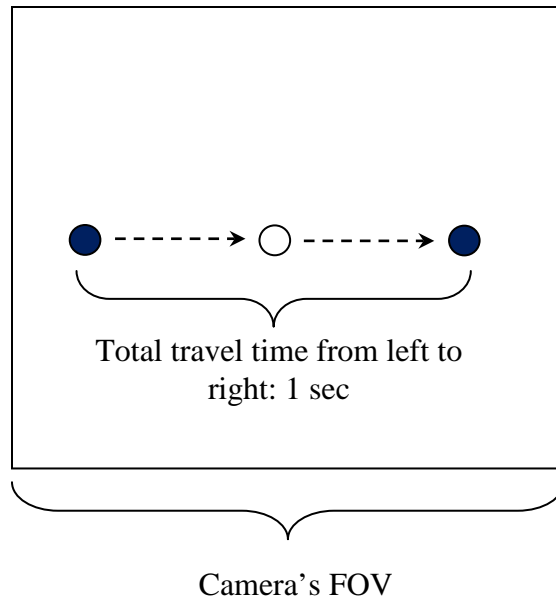


Figure 13. Target traversing at 33 pixels per second

Once the setup is complete, a recording of the target will be performed using the camera and the GUI. Each recording will last approximately 45 seconds and then stop recording for 30 seconds. For this test a series of 4 scenarios will be performed at different frame rates to determine if there is an observable delay in the image captured. Each scenario will be performed at least 15 times. Figure 15 flowcharts this process.

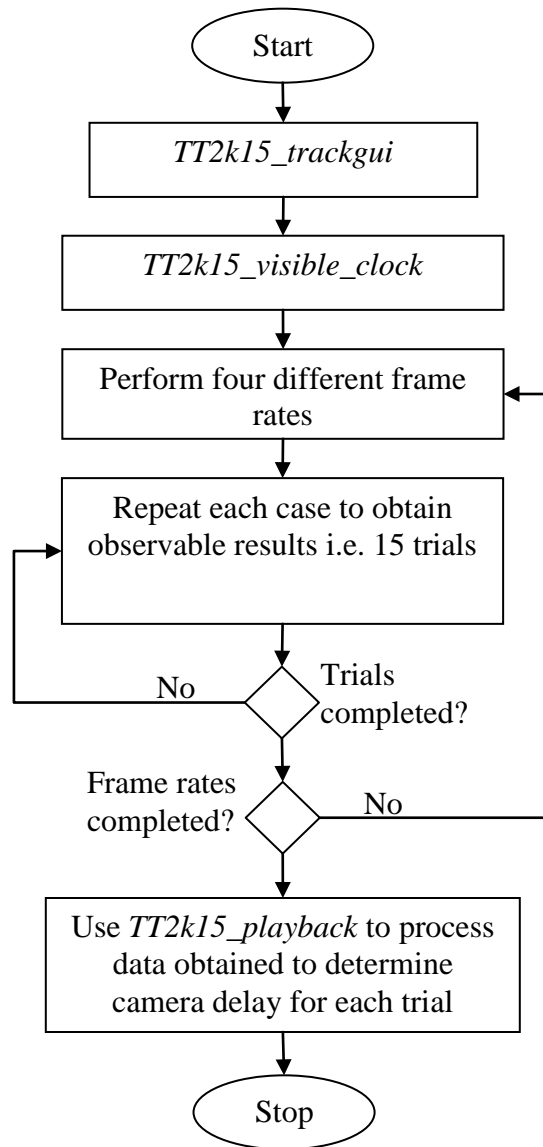


Figure 14. Flowchart of PC/MATLAB to Camera test

The *TT2k15_visible_clock* creates a simple white target on a black background. If a FOV is greater than the 0.4 degrees for the Philips webcam, then there are two options in order to change the displayed target to fit a different camera. To change the distance that the target travels, line 61 on the script must be modified.

```
'set(visclocktarget, "XData", 0.49 + (1 - 2*0.49)* mod(visclock(1,6),1)),',...
```

In this line the only value that needs to change is the decimal value. If the value is set at 0.5, then the target displayed will not move. For the Philips webcam the value is expected to be 0.48 or 0.49 due to the small 0.4 degrees of FOV. The second setting that can be modified is the target size. On line 102 of the script:

```
visclocktarget = plot(0.5,0.5,'ws','MarkerFaceColor','w','Parent',visclockaxes,'MarkerSize',1);
```

The MarkerSize setting '1' can be changed as desired. The target on the GUI must not be more than about ten pixels on the zoomed window as it will cover most of the area and the data collected may not be usable. For the Philips webcam it is expected to use MarkerSize 1 which is the smallest the GUI can generate. Figure 15 shows a bright object similar to the target expected which is about 10 pixels.

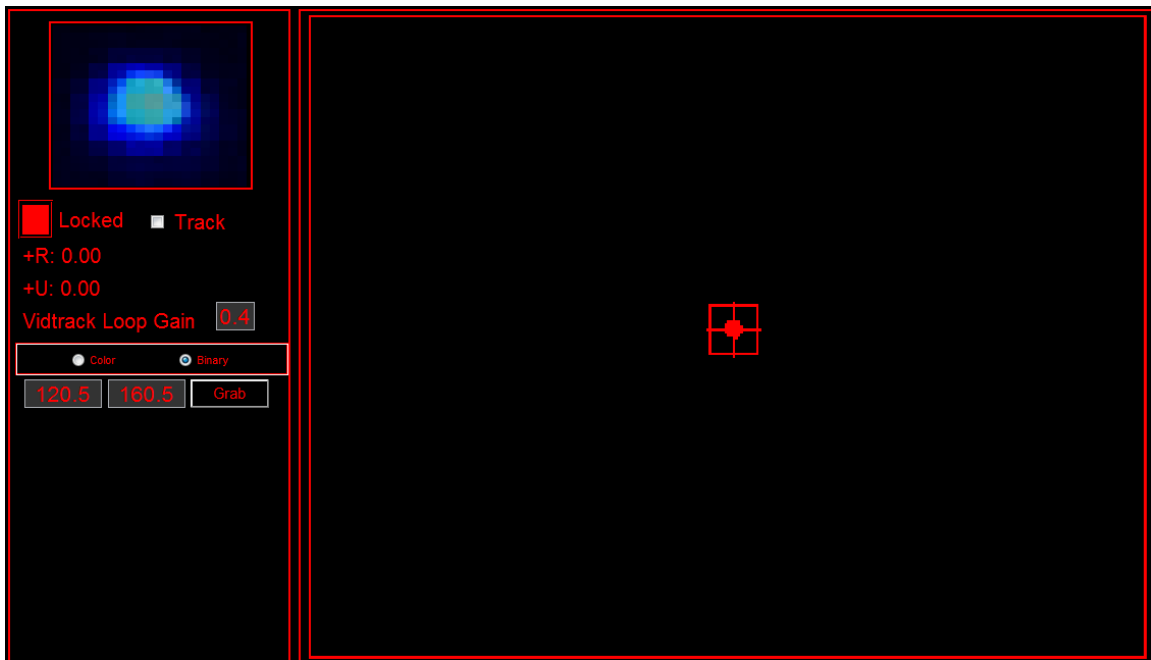


Figure 15. Desired target size for PC/MATLAB to camera

This process should culminate in the calibration of the camera and the information obtained will be used for the computation of the combined delay of all three components together. This script can be found in Appendix C. In order to determine this delay; this research will look into the files created during the recording. These files have an extension of .vid and .csv and are used with the *TT2k15_playback_v1_7* script created by Schmunk and Briggs [4, 10]. This script is useful because it is able to determine the number of frames captured by the camera. The script uses the information to fit a piecewise linear curve (since the target's original path is a sawtooth wave) within two standard deviations of the timing data which means that it is able to fit most of the data into a plot. Note that 2-sigma could be considered a "loose" fit, but is suitable for real-world applications. The data is then compared with the expected time when the target was created and the playback's algorithm is able to determine the delay between the static target and when it was captured by the camera.

PC/MATLAB, Meade mount and camera interface

Up to this point, a commanding time from the PC/MATLAB to the Meade mount has been determined, a camera calibration has been performed (PC/MATLAB to camera), and now the complete system will be tested as seen in Figure 9 to determine the delay between the three components. In order to determine this delay, a set of tests were performed. The basis for this test is to set a static target and "zero" the camera to have the target on the center of the GUI. From this known position the mount will follow a pre-determined path; for instance, left to right, forcing the static target out of the view of

the GUI and then back and through the target to the other side. The flowchart in Figure 16 shows the intended sequence of events.

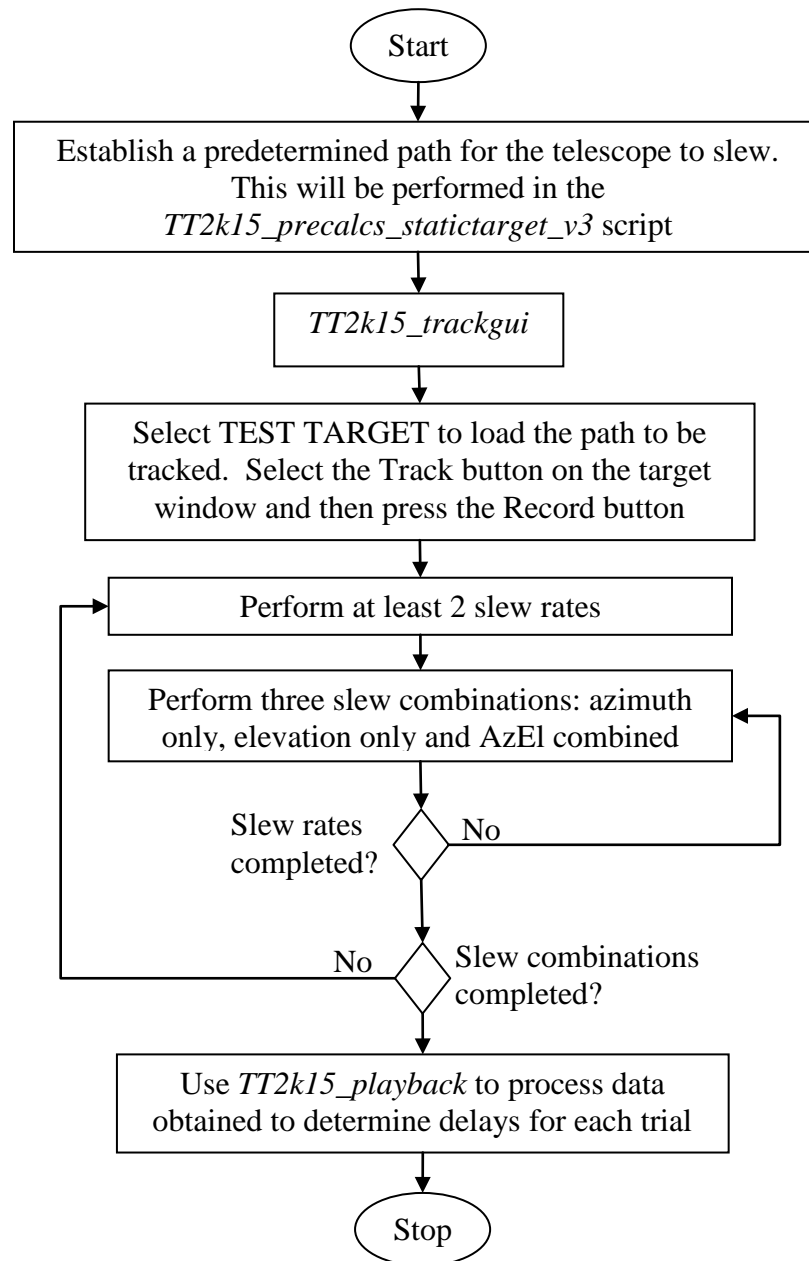


Figure 16. Flowchart of PC/MATLAB, camera and Meade mount test

The concept behind this test is to record this path and later analyze the data to determine the time it took for the mount to respond to commands initiated by the GUI. The data will also be used to determine when the GUI “thinks” it started to “see” the target. With this information it is possible to determine how far behind the mount is compared to the real position of the mount. This research will test azimuth only, elevation only and azimuth and elevation combined at the same period but different speeds to possibly determine maximum observable delays. It is predicted that the tests will not exceed the Meade mount maximum speed of 8 deg/sec [13] because the field of view of the system is 0.4×0.3 degrees. With the Orion telescope, this practically limits the testing speed to approximately 1.5 deg/sec, since it guarantees a few images will be collected on each sweep. A graphic representation of this setup is displayed in Figure 17

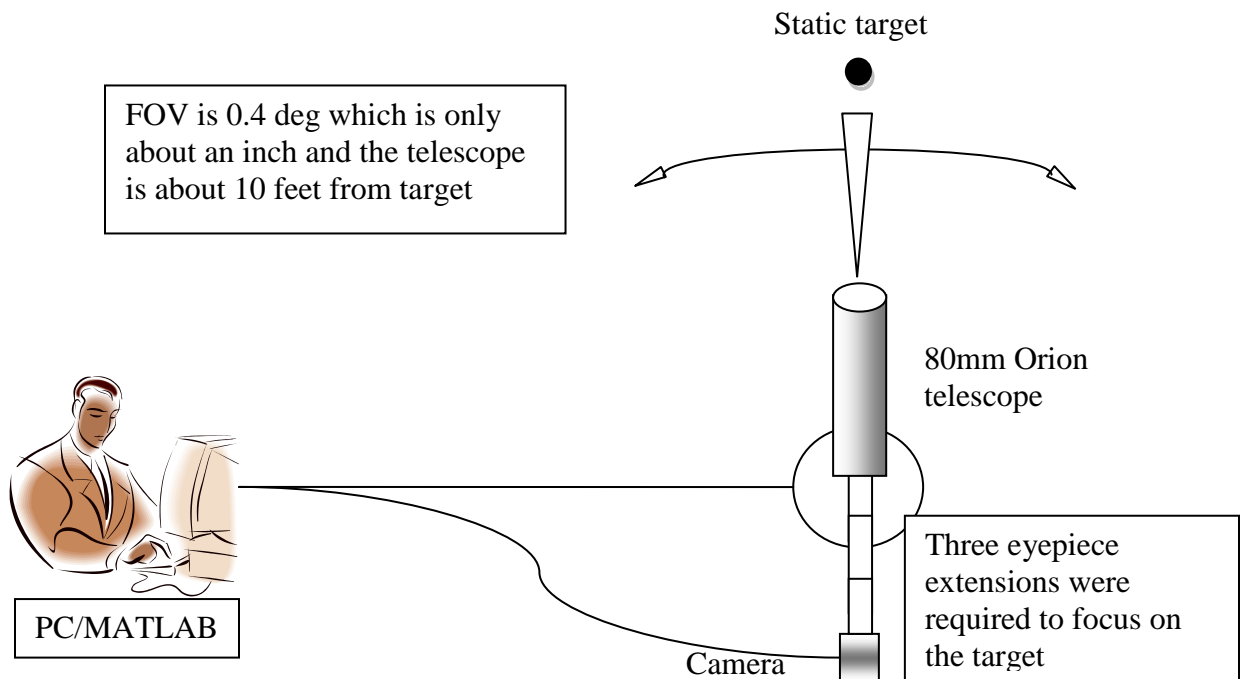


Figure 17. Limitation of target view due to NFOV

Similar to the PC/MATLAB to camera test, the method for analyzing the data obtained will be performed by using an updated version of the *TT2k15_playback_v2_3* which is able to determine the estimated time needed to be subtracted from the scope's position report to make it valid. This delay is a combined sum of delays between the PC/MATLAB to send the request, the time it takes the scope to process the request and the time of the position data inside the scope's computer at the time the data is sent back to the PC/MATLAB. This delay can be summarized as the cumulative delay between the PC/MATLAB, camera and Meade mount and is referred to as the *Backdelay*.

A second set of data extracted from the recorded video is the estimated time it takes for the telescope to start moving at the commanded rate which also includes the PC/MATLAB sub-delays, the time it takes the scope to process the request and the time it takes the drive to change its rate. This delay is referred to as the *Outdelay*. The *TT2k15_playback_v2_3* also provides a plot that shows several metrics to include the reported position of the scope, the estimated scope position, and a corrected path of the actual scope position. This plot also estimates when the static target crosses the center of the FOV based on a minimum of 5 targets being captured or recorded, but it also shows the estimated time when it crossed the center. Once the total mean delay is obtained, it will be used in the algorithm inside the GUI and the last procedure will be re-accomplished and results compared with the previous test to determine if the tracking controller displays observable improvements compared to the current tracking controller.

Summary

Starting with a higher-level system a decomposition of TeleTrak was performed starting with the OV-1 and SV-1. Using the OV-5 it was determined that a tracking subsystem could be improved. With the aid of the SV-6 it was discovered the potential locations that induce delays in the system. Understanding of the current configuration is paramount in the development of a new tracking system. Because this system is actively performing calculations in real time, it is imperative to systematically develop a set of instructions that will allow the approximation of cumulative delays in the system. The methodology presented can be re-accomplished on future system changes in hardware and software. The next chapter will include the findings of the proposed methodology used.

IV. Analysis and Results

Chapter Overview

Chapter III established the need for the development of a baseline in order to successfully lay the foundation for future students that may use or modify TeleTrak. This chapter will also present the findings of tests performed towards a better tracking controller for TeleTrak.

TeleTrak baseline

As previously described, the minimum hardware components used for this research are: main optic, telescope mount, PC with MATLAB, USB digital camera, USB cable, USB to serial cable, and power supply.

TT2k15_trackgui is the main file that allows for the successful operation of TeleTrak and is the core of the system. *TT2k15_trackgui* however is only the GUI, and in this section a list of required files will be presented in an effort to minimize confusion for future students. For an observation to take place there is a sequence of events that must be followed. A brief list can be found in Appendix F.

Before the start of any observation, pre-calculations of known RSO need to be accomplished to determine and establish known orbital information for later use in *TT2k15_trackgui*. The pre-calculations start from a single file *TT2k15_precalcs*, which uses the support of many other files. This list of files is found in Table 2.

Table 2. Files required for test setup

<i>TT2k15_precalcs.m</i>	
<i>calendar.m</i>	<i>jday.m</i>
<i>days2mdh.m</i>	<i>killer_getstars.m</i>
<i>dpper_vectorized.m</i>	<i>sgp4_vectorized.m</i>
<i>dspace_vectorized.m</i>	<i>sgp4init_vectorized.m</i>
<i>getLAST.m</i>	<i>TT2k15_getcatalogsats.m</i>
<i>getsite.m</i>	<i>twoline2rv_simple.m</i>
<i>getsun.m</i>	<i>watcher_getstarazel.m</i>
<i>getzenith.m</i>	<i>getdarkness.m</i>

TT2k15_precalcs.m is used to generate orbit predictions in azimuth and elevation.

Table 2 contains the required MATLAB files to perform the required pre-calculations necessary for the observation of known space objects by using the TLE list. As previously described, the TLE list can be obtained via the space-track website at www.spacetrack.org and procedures on how to obtain it are included in Appendix D.

After the pre-calculations have been completed, the next step on the day in the life of TeleTrak is to setup the equipment for observation. The main files required for the basic TeleTrak system are shown in Table 3. As mentioned before, *TT2k15_trackgui* is not the main file. It is however the file that the user runs as it provides the user interface.

In this list some files are directly used by the GUI while others are functions called within other functions. Additionally, the .mat files contain values of variables created by other programs for instance the *TT2k15_precalcs.m* creates the *precalc_results.mat* which are then loaded by *TT2k15_trackgui*. For ease of readability the files have been divided according to their extensions. All of the .m files are in Table 3 and the rest of the files are in Table 4.

Table 3. Required files for GUI operation

<i>TT2k15_trackgui.m</i> is the starting file of the GUI	
<i>briggs_diffvector_watcher.m</i>	<i>tiptilt_truetorawrate.m</i>
<i>bwlabel_coarse_v2.m</i>	<i>traknet_buildsattract.m</i>
<i>calendar.m</i>	<i>traknet_createcammenu.m</i>
<i>degrees2dms.m</i>	<i>traknet_createschedule.m</i>
<i>diffvector_prototype.m</i>	<i>traknet_createstarmap.m</i>
<i>dscom.m</i>	<i>traknet_createstarmenu.m</i>
<i>dsinit.m</i>	<i>traknet_refreshgui.m</i>
<i>findtelescope.m</i>	<i>traknet_scopecontroller.m</i>
<i>fov_to_azel.m</i>	<i>traknet_updatetargetstate.m</i>
<i>getgravc.m</i>	<i>traknet_zoompainter.m</i>
<i>getsun.m</i>	<i>TT2k15_bwlabel_coarse_v2.m</i>
<i>gpssync.m</i>	<i>TT2k15_initcamera.m</i>
<i>gstime.m</i>	<i>TT2k15_initscope.m</i>
<i>hms2deg.m</i>	<i>TT2k15_save_scopestate.m</i>
<i>initl.m</i>	<i>TT2k15_streamer.m</i>
<i>jday_clock.m</i>	<i>UFB_pucktracker.m</i>
<i>killer_getstarazel.m</i>	<i>watcher_streak2SEZ.m</i>
<i>meadestring_to_angle.m</i>	<i>watcher_streak2URD_v2.m</i>
<i>newcamera.m</i>	<i>watcher_trackpainter.m</i>

Table 4. Support files to TeleTrak

<i>packer_v4.c</i>	<i>ha.txt</i>
<i>packer_v4.mexw64</i>	<i>logfile.txt</i>
<i>qs.mag</i>	<i>precalc_tle.txt</i>
<i>camconfig.mat</i>	<i>specials.txt</i>
<i>circ179_nutation_terms.mat</i>	<i>sun.txt</i>
<i>precalc_results.mat</i>	<i>mcnames</i>
<i>tiptilt.mat</i>	<i>catalog.dat</i>

After the observation has taken place the files with the recorded video can be processed using the *TT2k15_playback_v2_3* file. In this case *TT2k15_playback_v2_3* is the main file and utilizes the all of the associated files as seen on Table 5. Besides these

files the *TT2k15_playback_v2_3* will request to load the .vid and .csv files from the “\Videos” folder within the main folder.

Table 5. Required files for post analysis

<i>TT2k15_playback_v2_3.m</i> is the main file for processing .vid and .csv data	
<i>diffvector_prototype.m</i>	<i>TT2k15_load_scopestate.m</i>
<i>INVJDAY.m</i>	<i>UFB_pucktracker.m</i>
<i>jday_clock.m</i>	<i>watcher_streak2URD_v2.m</i>
<i>rate_integrator_avg_v7.m</i>	<i>watcher_trackpainter.m</i>
<i>rate_integrator_avg_v7_notiptiltfast.m</i>	<i>unpacker_v4.c</i>
<i>tiptilt_rawtotrue.m</i>	<i>unpacker_v4.mexw64</i>

Although the files described are the minimum, additional files were found in the TeleTrak folder. Upon research and communicating with SMEs it was determined that the files in Table 6 are not in use or are currently incompatible with the single system. These files have been kept for archival purposes and also to provide continuity for future research as they can be combined with previous work and may be used to restore previous capabilities like network supportability.

Table 6. Files not currently in use (requires validation before use)

Files not currently in use	
<i>buffer_init.m</i>	developed for draft network-operations mode, not "core"
<i>check_star_interps.m</i>	test script, not required for ops
<i>converting_mount_tilt.m</i>	test script, replaced by draft tiptilt_radial_error, which currently doesn't work so great
<i>sendcommand.m</i>	a convenience file
<i>swapscreen.m</i>	used for draft network operations only
<i>tiptilt_radial_createsurface.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>tiptilt_radial_error.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_buildglobelos.m</i>	developed for draft network-operations mode, not "core"
<i>traknet_buildglobetrack.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_close.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_commandgui.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_init_echo.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_open.m</i>	unvalidated approach to calculating tiptilt "realtime," has issues currently
<i>traknet_playback_avimaker.m</i>	no longer usable w/current code base
<i>traknet_trackgui_align.m</i>	no longer usable w/current code base
<i>TT2k15_azerrorfunction.m</i>	unvalidated alternate approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>TT2k15_correctionmapbuilder.m</i>	initial draft of unvalidated alternate approach to calculating tiptilt "realtime."
<i>TT2k15_correctionmapbuilder_staticazbias.m</i>	improved draft of unvalidated alternate approach to calculating tiptilt "realtime."
<i>TT2k15_createsurfacefit_az.m</i>	unvalidated alternate approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working

<i>TT2k15_createsurfacefit_el.m</i>	unvalidated alternate approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>TT2k15_createsurfacefit_rms.m</i>	unvalidated but best-performing approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>TT2k15_createsurfacefit_rms_v2.m</i>	unvalidated but best-performing approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>TT2k15_elerrorfunction.m</i>	unvalidated alternate approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>TT2k15_rmerrorfunction_v2.m</i>	unvalidated but best-performing approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>watcher_orbit_estimator_v3.m</i>	used only for Briggs' "staring mode," currently un-re-validated and probably suspect
<i>watcher_orbit_propagator.m</i>	unvalidated but best-performing approach to calculating tiptilt "realtime," will likely be removed once tiptilt_radial_error is working
<i>camconfig_old.mat</i>	backup
<i>QUICKSAT.DAT</i>	test file, used for results with comparison with another satellite tracking program

PC/MATLAB to Meade mount results

For this test, the Meade mount was connected to the PC/MATLAB directly using a serial to USB converter. –As described before, each case would be tested 40 times as it was possible to obtain observable results. In this case the “period” is a global variable name used in the *serial_tester_v3* script as the time span to execute the code; therefore, at a period of one second the script would repeat every second which then would compute the time it took from requesting a pointing angle from the telescope mount to the time the telescope replied. From the data collected, the mean delay time it took for each case as

well as the standard deviation was computed. Results from the different cases can be observed in Table 7.

Table 7. Mean and standard deviation at different periods

Period (sec)	Mean delay (sec)		Std Dev (sec)	
	Static	Dynamic	Static	Dynamic
2	0.42174	0.416136	0.006036	0.011001
1	0.42131	0.418982	0.005514	0.008463
0.5	0.229615	0.228102	0.195413	0.201177
0.4	0.031751	0.037038	0.015511	0.017493
0.3	0.102281	0.101952	0.044516	0.048255
0.2	0.03744	0.040582	0.022181	0.029174
0.1	0.074692	0.077289	0.086252	0.088106

Initial observation of the data collected may lead to an assertion that at a period of 0.5 seconds, the telescope had a greater standard deviation than the rest of the cases. It also appeared that with a “period” of 0.4 and 0.2 there was no significant delay difference. Because of this, it became important to try to plot the telescope’s response. Figure 18-24 show the response between the PC/MATLAB and the Meade mount. It was observed that for a 1 second period and greater, the Meade mount was capable of performing the task. However, if the space object to be tracked is in a LEO orbit then the Meade mount may not be able to track the object as it may transit out of the FOV between updates. For a typical LEO object that is directly overhead it takes approximately 9 minutes to travel from rise to set ($\approx 180^\circ/9 \text{ min}$). This means that a typical LEO object travels at about 1/3 of degree per second. Since the FOV of the 80mm Orion optic is 0.4 degrees across or 0.2 degrees from the center of the FOV, it suggests that updating tracking movements every second is not feasible.

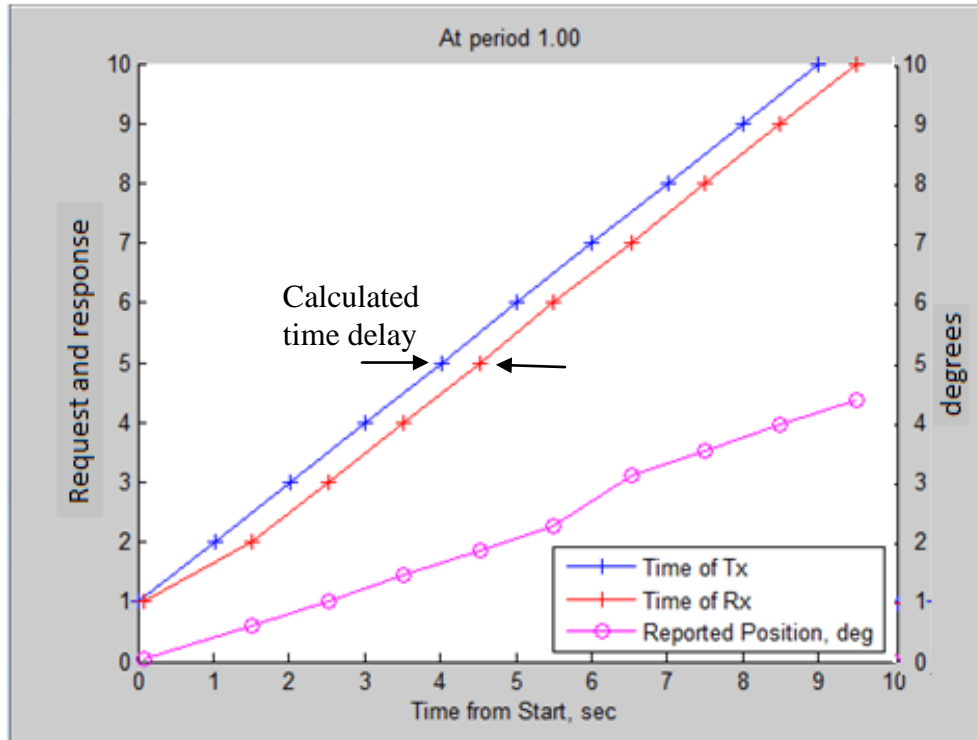


Figure 18. PC/MATLAB to Meade mount response at period = 1.0

At a period of 0.5 second it is observed that the telescope is able to report back to the PC/MATLAB the pointing position without repeated values. However, at a period of 0.1 second for instance, it can be observed in Figure 20 that the time from the request to move until the time it took the Meade mount to respond was not consistent. In Figure 20, the gap between the Time of Tx and the Time of Rx varies. Also, the reported position is not consistent which implies that perhaps the Meade mount is being over tasked. As is observed, the reported position sometimes is the same as the previous value and sometimes it changes erratically.

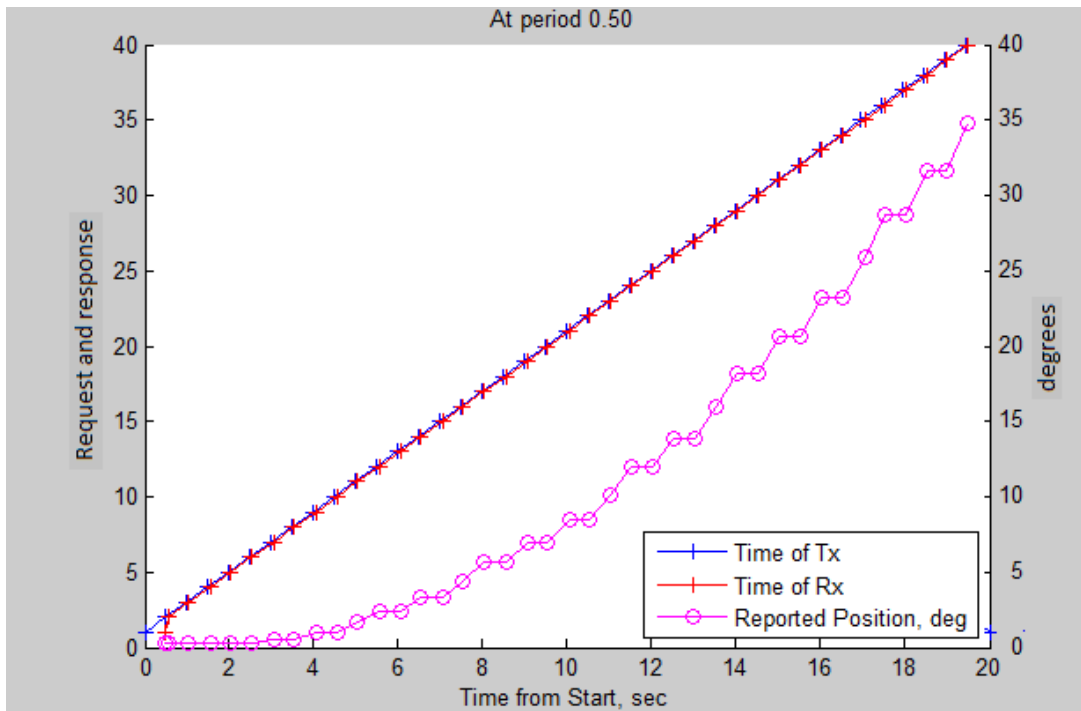


Figure 19. PC/MATLAB to Meade mount response at period = 0.5

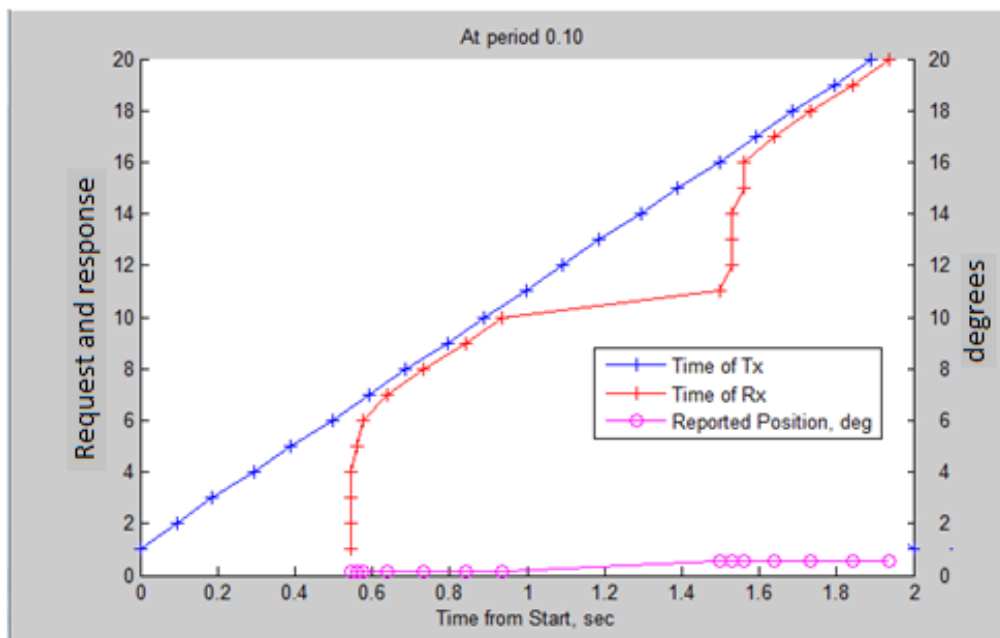


Figure 20. PC/MATLAB to Meade mount response at period = 0.1

Through the previous test it was found that the rate of tasking the mount has a great impact on how accurate the information of the position of the telescope is. Although asking the position of the telescope once every 2 seconds appeared to be the most accurate it also seemed that the mount would go on “standby” mode meaning it would not necessarily update its position. Long periods between requests could affect the existing algorithms’ ability to estimate the telescope’s “current” position, which impacts the overall performance. On the other hand, tasking the mount every tenth of a second proved to be too much for the mount as it appears that even if the mount is active and moving, the information within the mount is not updated as often, meaning that perhaps the mount would keep the information in the buffer if it appeared to be in the same location. This was considered as over tasking the mount, with no obvious loss of capability for most satellite tracking applications. Therefore, it was determined that asking the position of the telescope at a half a second interval (a setting which was used since 2006) gave the best and most accurate information shown in Figure 19. It seemed that the mount was responsive enough to provide almost a new position every time it was requested. The use of this information will become apparent later when combining all the components.

PC/MATLAB to camera results

For this test the 80mm Orion optic had the SPC 900NC PC camera attached. The camera was connected to the PC/MATLAB via USB. The Orion optic was pointed at a second monitor where the PC/MATLAB resided. As described before, the Orion optic remained in one place while the target on the monitor moved from left to right.

Recording of the target was done by the GUI through the camera. Each trial would have a 45 second run with a 45 second downtime to prevent biasing the test. For the test, the serial COM port was utilized to “detect” the telescope however no pointing commands were sent to the Meade mount. After one set of trials was completed the data could be processed using the “playback” script. Below is a walkthrough of processing a single trial.

Figure 21 shows one frame of the recorded output. The crosshair marks the center of the target that sweeps across the screen. If more than one target is captured on the same frame due to the length of the exposure, then the rightmost object is selected.

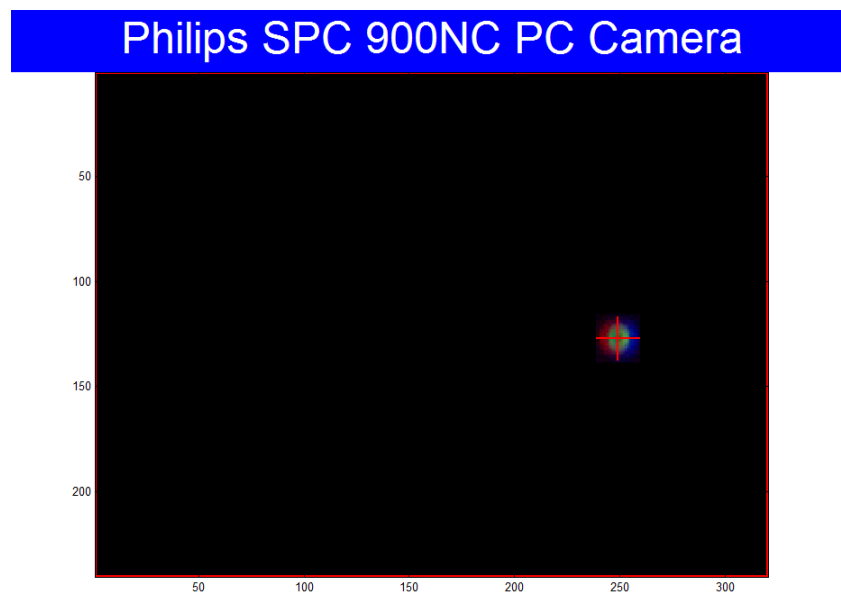


Figure 21. Target displayed on single frame during playback

Next, when the playback script is set to *Camera Cal* mode, it assumes that the camera filmed a true sawtooth. This sawtooth is generated with a period of one second and amplitude that matches the size of the filmed sweep. The amplitude is calculated by

subtracting the maximum value of all detected dots from the minimum of all detected dots. The playback script removes outliers to make the data more cohesive. Once the curve fit is established, the delay is computed by comparing where the peak value of the sawtooth falls with respect to the closest integer second. The computed delay is the result of the raw data on a sawtooth fit curve and the red line is the nearest integer as the original target moves at exactly one second per cycle. A full explanation provided from Schmunk via correspondence can be found in Appendix G. A sample of the obtained delay is shown in Figure 22.

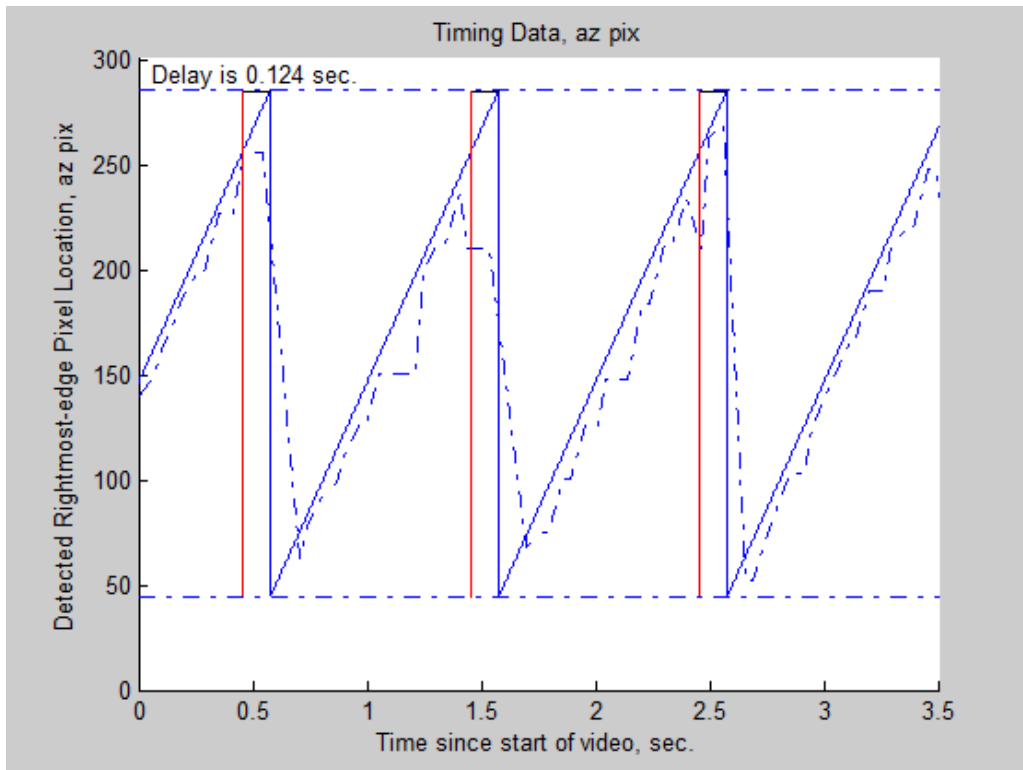


Figure 22. Computed delay using a sawtooth curve fit

For each frame rate a test was performed 15 times and then the mean and standard deviation was computed. Table 8 shows the camera resolution and frame rates used to

perform the different tests as well as the computed mean and standard deviation of the results.

Table 8. Computed mean and standard deviation of the PC/MATLAB to camera delay

The resolution used for the experiment was 320x240		
Frames per second	Mean (sec)	Std Dev (sec)
20	0.124467	0.012
15	0.129643	0.016
10	0.168929	0.017
5	0.275214	0.021

The plots in Figure 23 and Figure 24 show that at lower frames per second, the processing of the data collected increases because each exposure is longer which increases the delay. It was also observed that the standard deviation is very small which indicates high confidence of reproducing similar results. It is important to note that a frame rate greater than 20 is not recommended because small objects require longer exposure to be able to collect enough photons to be able to get any useful information for characterization.

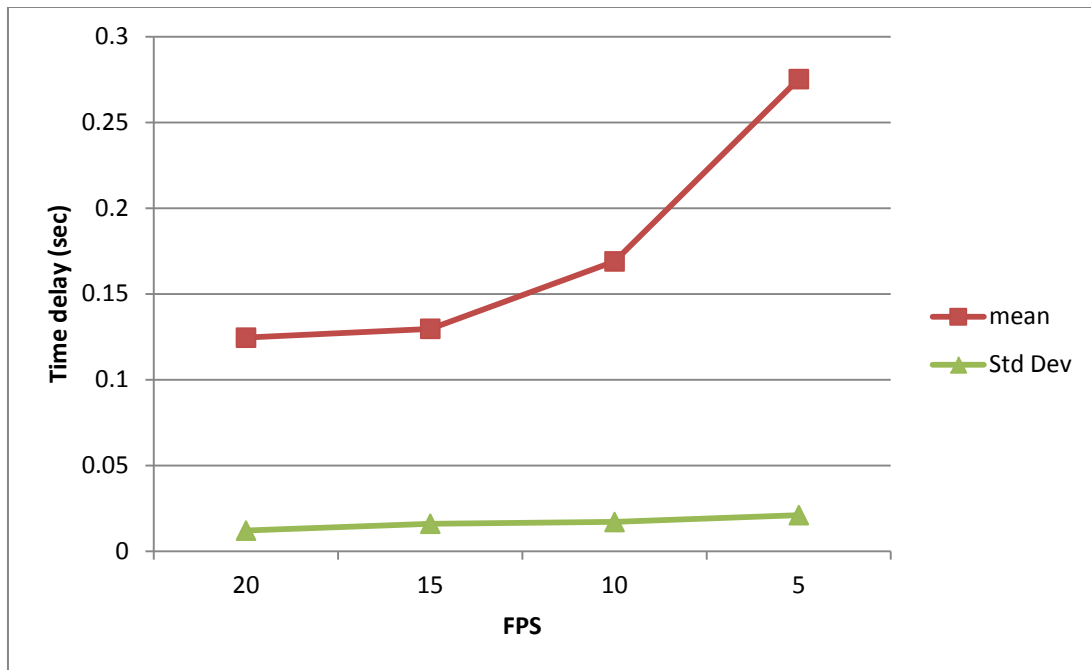


Figure 23. Average and standard deviation of PC/MATLAB to camera delay

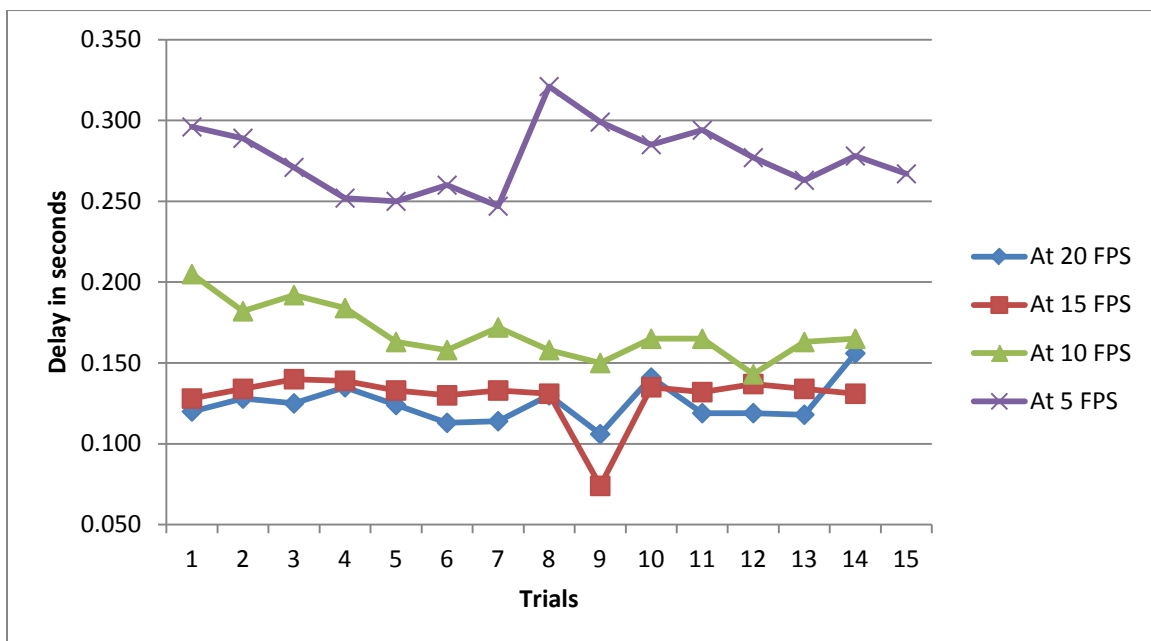


Figure 24. Camera delay at different frames per second rates

PC/MATLAB, Meade mount and camera interface results

This test combines all three components that directly affect the tracking of space objects. For this procedure a static target was placed on a tripod on one side of the room in the ASOC and the telescope was placed on the other side of the room similar to Figure 17. This was done to be able to focus on the target. Although several options for light source of the target were considered, it was determined that an LED would be the best fit for this test. Reflecting light on a white spot on the wall with a black background using LED or a red laser as a light source proved difficult. Using the reflection of the LED was not able to provide a clear and bright target and the laser proved to be too bright which would saturate the camera creating multiple targets for which was useless as only one target was needed. Pointing a laser light source directly to the camera was determined to be dangerous as it could damage the camera [14]. Limitations with this test were the length of the cable as both the camera and Meade mount had to reach the computer with the GUI and also the test had to be performed in a “dark” environment as the algorithm was written to detect bright objects. As it can be observed, in Figure 25 and Figure 26, the FOV is very small and the telescope mount could not go fast otherwise it would never detect the LED.

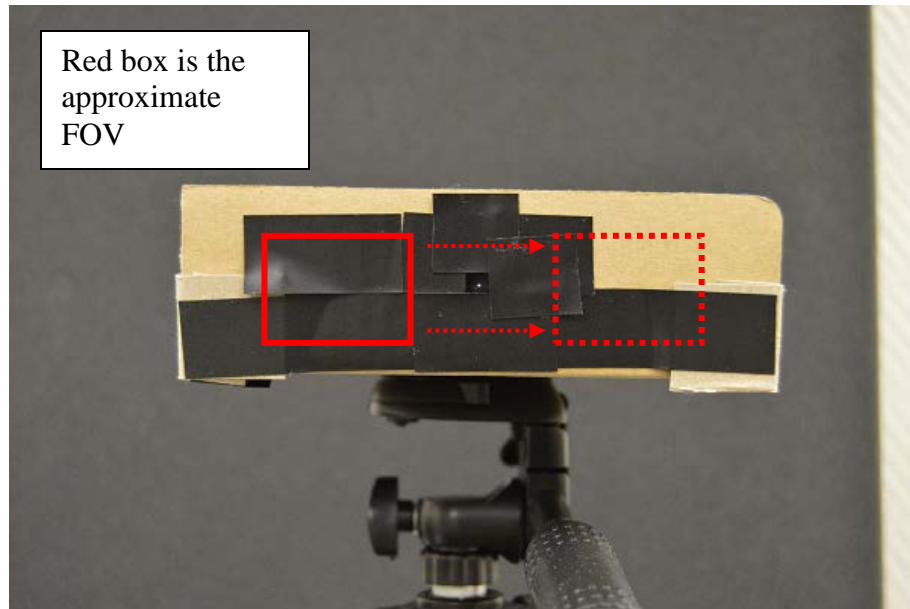


Figure 25. Static LED target

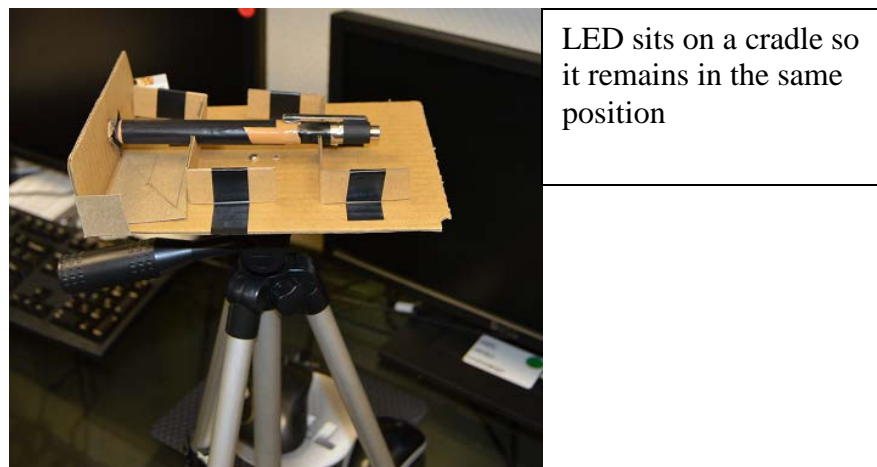


Figure 26. LED on tripod platform

As previously mentioned, it is desired to determine the Backdelay and Outdelay of the controller. The method used to determine this delay was to slew the Meade mount at different rates and compute the mean and standard deviation of these delays. Using the *TT2k15_precalcs_statictarget*, a predetermined path for the Meade mount to follow at a

determined slew rate was created. For this test, the “period” is still a variable within the GUI; however, in Figure 27 shows that the “period” of 10 seconds results in a clipped sinusoid with a 16 second period. In order to allow for the Meade mount to “settle” a 3 second pause was created at the end of each slew of the telescope to create a smooth transition for the Meade mount to move to the new direction. Results from one instance of azimuth test at 1.29 half amplitude and 0.8 deg/sec can be seen in Figure 27.

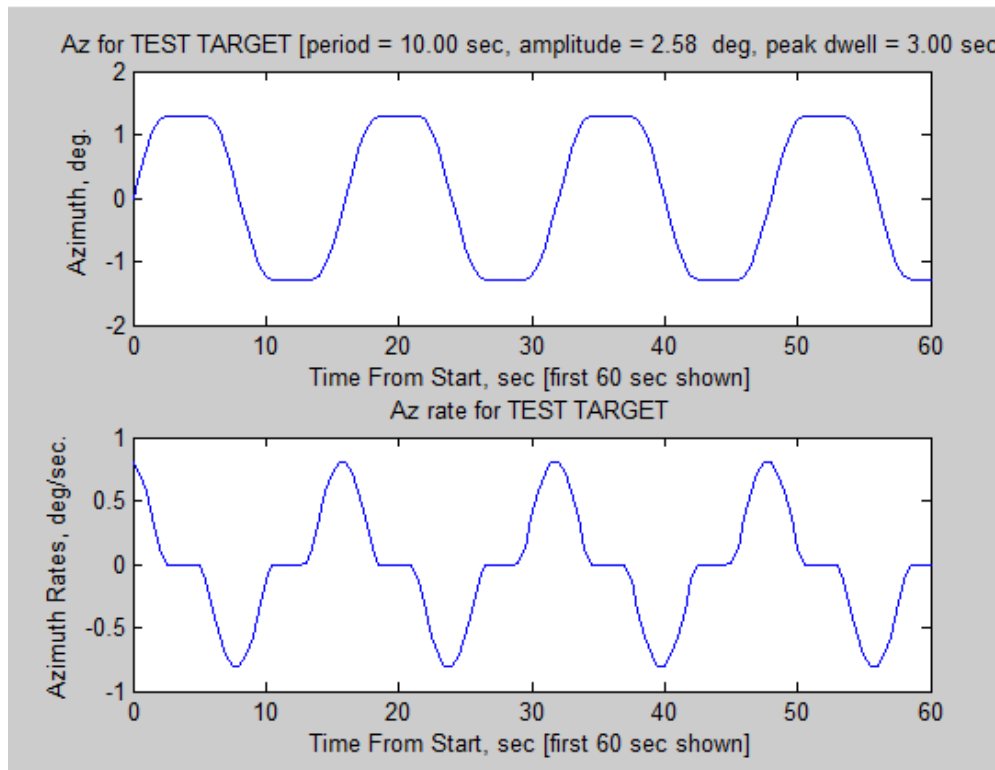


Figure 27. Azimuth input generated with the *TT2k15_precalcs_statictarget* file

The following information is used for constructing Figure 28:

- Black is the pattern that shows the telescope's reported position
- Red is the GUI's position request to the telescope's mount
- Solid blue displays that the target is not in view or is not detected

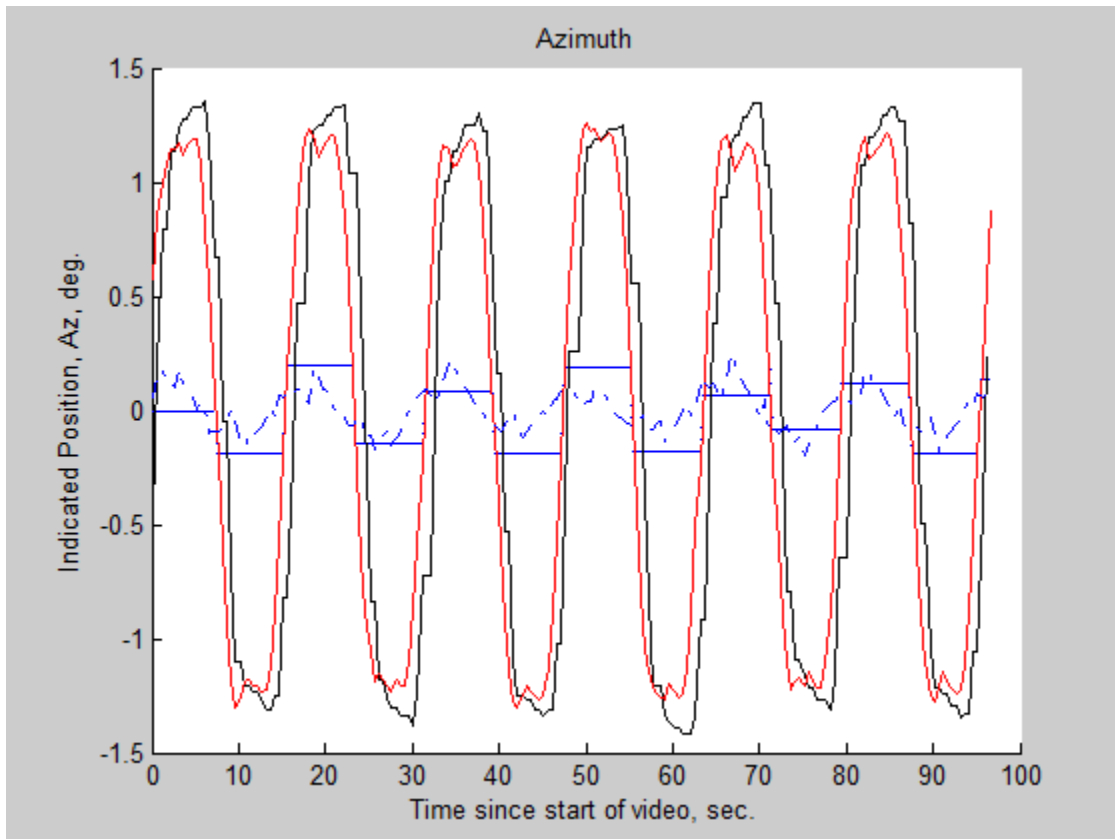


Figure 28. Sample of azimuth data obtained from single instance (0.8 deg/sec)

After the script completes the computations it displays (as in Figure 29) the best fit plot that provides the following information:

- Magenta represents the calculated “best fit” of where the telescope “should be” if the correct delay correction is applied.
- Red squares represent when the static target was first observed by the camera but adjusted as the camera’s delay influences the actual position of the mount.
- Blue squares represent the “corrected” position where the object could have been first observed and it demonstrates that the algorithm is able to determine the backdelay as well as the outdelay

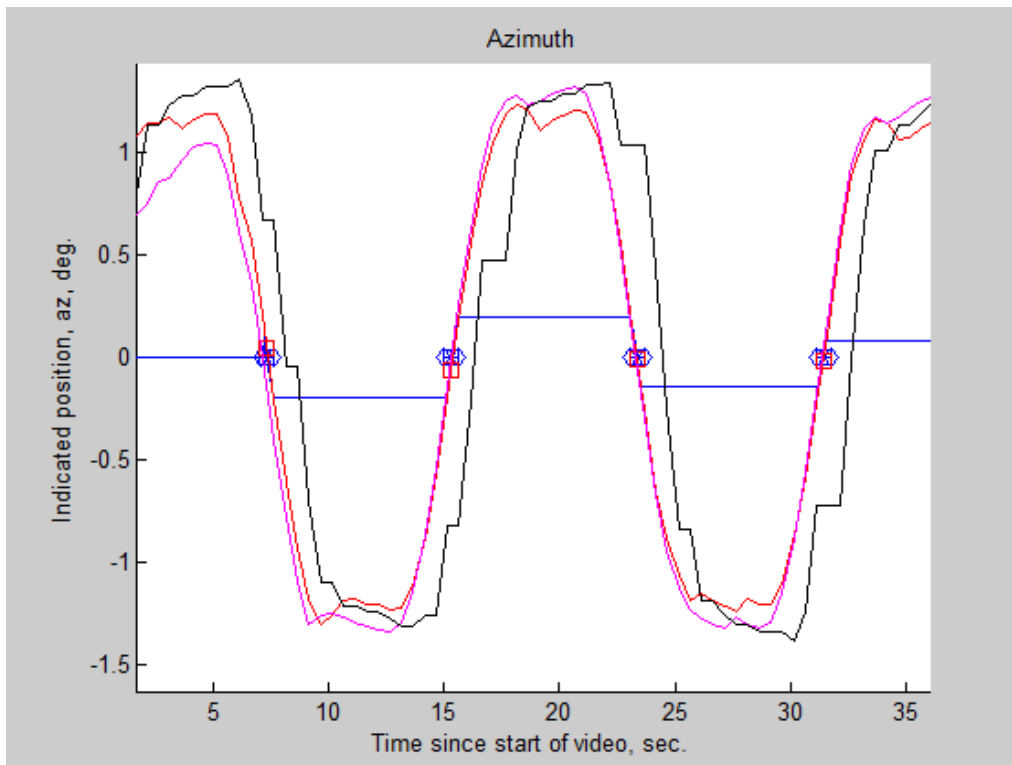


Figure 29. Azimuth comparisons including best fit

The test was performed with up to three different slew rates but with the same period of 10 seconds. Each case was performed 10 times for consistency and the results are provided below:

Table 9. Mean and standard deviation of the Backdelay and Outdelay

Azimuth				
Meade mount rate (deg/sec)	Backdelay mean (sec)	Outdelay mean (sec)	Backdelay std dev (sec)	Outdelay std dev (sec)
0.5	0.506385	0.283231	0.007643	0.048293
1	0.673417	0.221333	0.031012	0.022765
1.5	0.690444	0.199444	0.032871	0.021448
Elevation				
0.5	0.522800	0.337300	0.147571	0.029687
1	0.574455	0.287909	0.045114	0.023248
Azimuth and elevation combined				
0.5	0.515727	0.333864	0.040974	0.046357
0.8	0.608545	0.253136	0.069042	0.035937
1	0.666611	0.254389	0.075740	0.036886

It can be observed that for every case the backdelay is greater than the outdelay.

As explained earlier, this is expected because this delay is the combined delay of the tracking system whereas outdelay is only the time that it takes the telescope to start moving and is expected to be smaller. This is because most of the delay appears to reside in the Meade mount. From Table 9 and Figure 30 – 36 it can also be observed that the standard deviation is small which again gives the confidence that the tests are repeatable.

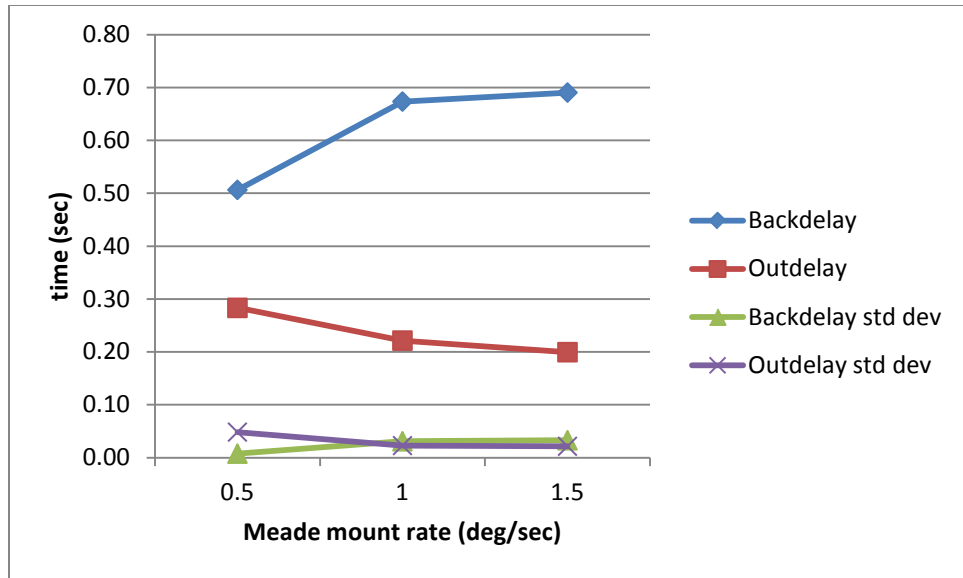


Figure 30. Azimuth delays

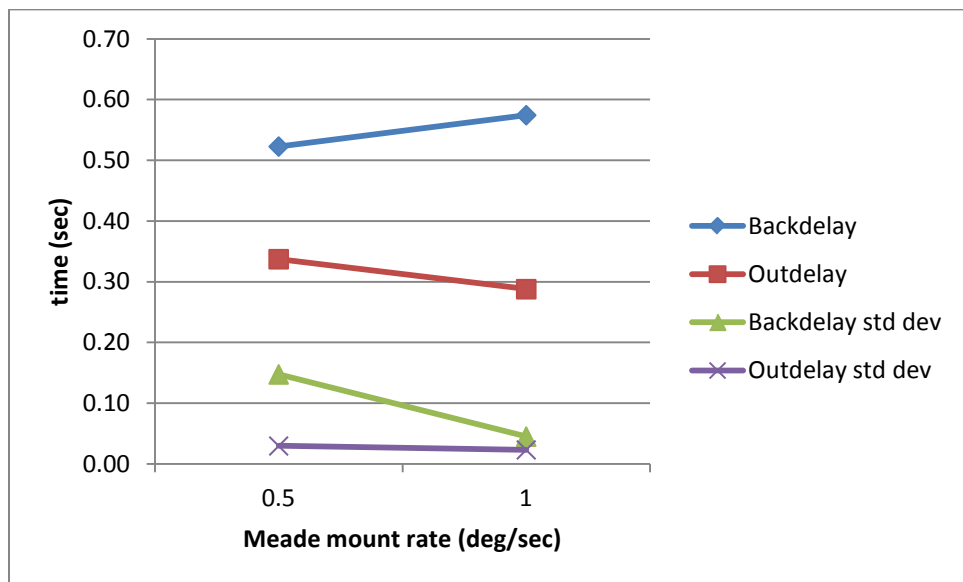


Figure 31. Elevation delays

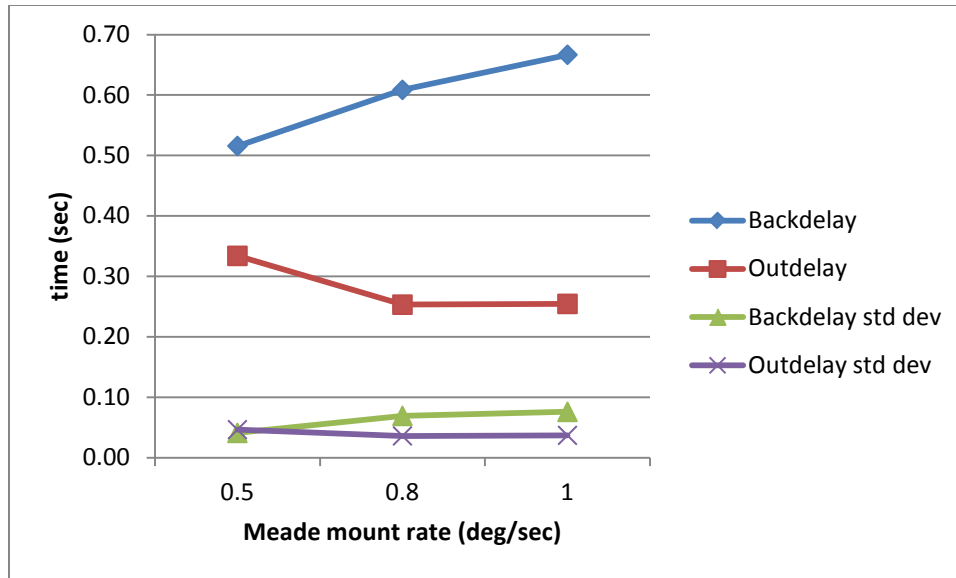


Figure 32. Azimuth and Elevation delays

Summary

Through this process, these test results revealed both expected and unexpected behaviors, which can be used to drive further development and refinement.

Here is a summary of findings:

- Partially-expected results:
 - The “backdelay” value, which is an estimate of when the telescope's position is measured compared to when it is requested, was larger than expected (Schmunk had speculated it was “wrong” but unclear on its magnitude or origin). Previous estimates had placed it at 0.2 seconds. The now-estimated value of 0.59 seconds suggests that earlier testing methods might have overlooked data arriving over one period late. Since requests are sent every 0.5 seconds, an entire period was erroneously subtracted from the estimate.
 - The “outdelay” value, which is an estimate of when the telescope responds to a command compared to when it is sent, was in-family with previous estimates (0.36 seconds, versus 0.275 seconds now).
- Unexpected results:

- On occasion, the telescope appeared to “drop” a request for position, this was definitely observed in elevation. Since elevation is requested shortly after azimuth, the telescope is likely the source of this anomaly, since it is unable to process closely-spaced commands. Meade's 2002 interface document says their telescopes should respond with a “busy” signal (ASCII NAK), but this was never observed at AFIT. Other users have noted the same [2, 6]. This same source describes an 80-character max for the input buffer; TeleTrak currently does not send this many characters in short bursts but it's possible that this explains the behavior (since the rate at which commands are removed from the buffer is not precisely known). Meade's 2010 interface document does not address the ASCII NAK nor the buffer size but recent findings suggest that it still something to consider.
- The code, as written, doesn't account for this possibility and when a dropped request occurs, it causes the timestamp of position reports to be shifted by one period from the dropped request time forward. Therefore, tracking degrades as the time of the track increases which must be accounted for.

V. Conclusions and Recommendations

Conclusions of Research

This research used a subset of the SE process to focus on the requirements for TeleTrak. The combined use of operational and system views were instrumental in developing the testing conditions in this research. Furthermore, the use of the activity diagram and sequence diagrams assisted in the development of an understanding of the tracking subsystem. Once the basic understanding of the tracking subsystem was obtained, this research developed a test processes to determine if the tracking subsystem could be improved. However, it has been determined that previous TeleTrak designs are basically sound, and with further refinement and calibration can be expected to support a wide range of satellite tracking research.

At the beginning of this research effort, TeleTrak was inoperable but with some adjustments the GUI has been successfully utilized establishing a new baseline for future research topics. Schmunk and Briggs were very successful at determining the delay of the system. By utilizing a systematic approach it has been determined that with this particular configuration, the delay between the camera, telescope mount and MATLAB is approximately 0.59 seconds which can now be implemented in the GUI to better utilize its algorithm in determining the trajectory of the object in view.

In addition to setting a baseline for TeleTrak, this research also re-affirmed limitations of the system that were suspected but not proven.

Recommendations

Recommendations for follow-on research are based on the results of this research.

- Continue efforts to systematically develop a plan to test and operate TeleTrak.
- Re-establish network capabilities. It has been proven that TeleTrak should be capable of being fully operational from the AFIT Space Operations Center (ASOC) using a Virtual Private Network (VPN).
- Incorporate multiple telescopes for observation. Once TeleTrak is fully networked, it can be used to tip and cue as AFIT currently has two full sets of telescopes and will add/replace them with newer equipment. A draft version of a non-VPN-based set of code exists which would be far superior for these applications. Now that the individual-telescope interfaces are refined, that code should continue with development.

Summary

Efforts done by prior researchers created a complete showcase for the use of COTS equipment in real-world applications. This project shows that COTS remains viable for many subsystems, but it also highlights that getting maximum performance requires semi-specialized coding, testing, and know-how. Successful application of SE operational and system views makes this possible, and of all the systems engineering approaches, spiral appears to work best for projects like TeleTrak. This project re-established documentation and understanding for future AFIT graduate students, and leaves behind some useful tools and concepts for not only the explicit equipment used in the project, but for other COTS-based telescope systems that may be used in the future.

Appendix A. Multi-TeleTrakNet Concept of Operations

Developed: 30 August 2011, Updated: 14 February 2012 by Shreiner [3]

Introduction. The following is the Concept of Operations (CONOPS) for a multiple site system of Air Force Institute of Technology (AFIT) commercial off the shelf (COTS) telescope networked systems (TeleTrakNet) supporting the Space Surveillance Network (SSN) as a contributing sensor. This larger network of sites will be referred to as Multi-TeleTrakNet to limit any confusion.

1. **Purpose.** The purpose of this CONOPS is to determine and then exploit the potential opportunity a Multi-TeleTrakNet system could provide to the SSN.

2. Time Horizon, Assumptions and Risks.

2.1 Time Horizon. The time horizon for the system to be operating is in the time frame of two to three years after 1 January 2012.

2.2 Assumptions. The following assumptions of the CONOPS are made:

- The Multi-TeleTrakNet system helps solve the problem and assumption that the number of objects in Low Earth Orbit (LEO) will continue to grow while the size of objects will decrease requiring higher capable systems by allowing the Multi-TeleTrakNet system handle lower and larger priority objects.
- The current AFIT TeleTrakNet system will continue to have students, faculty supporting the system for research, analysis, and support.
- The Global Information Grid (GIG) is operational.

2.3 Risks. Associated risks to Multi-TeleTrakNet system or ability to perform its mission are:

- Decreasing budgets and personnel may force the system into longer downtimes which will impact updates to the JSPOC satellite catalog.
- With enough sites the possibility for large generation of data is possible forcing an overloading of current command and control (C2) computers.

3. **Description of the Military Challenges.** The description of military challenges for the Multi-TeleTrakNet system mirrored the military challenges proposed by the *Operating Concept of the Space Surveillance Telescope (SST)*.

3.1 Adversary space capabilities are rapidly and significantly increasing, stressing SSA resources that are already constrained by development timelines, costs and operational locations. The number of low Earth orbit (LEO), medium Earth orbit (MEO) and geostationary Earth orbit (GEO) satellites will increase as more countries increase their space capabilities through indigenous development or procurement through third party vendors. The growth in countries using space for communications, Positioning, Navigation and Timing, and ISR&E monitoring has significantly increased the number of space related ground stations (data, telemetry, tracking, and commanding);

communications links; and resident space objects. Additional Earth-orbiting space platforms increase the challenge to maintain situational awareness of the space domain.

3.2 Technology miniaturization continues, creating smaller payloads and satellites. Micro-electro-mechanical systems and nanotechnology systems have improved and have been incorporated in satellite systems, creating smaller and cheaper satellites, and reducing launch costs. This development stresses space surveillance capabilities to detect, track and monitor objects in support of SSA. Satellite fabrication technologies have also challenged the ability to provide SSA. The use of non-glinting material, radar absorbing material and using diffusing angles in a spacecraft's design continues to reduce spacecraft detectability. Launches carrying multiple payloads stress SSA capabilities when combined with the above spacecraft applications.

3.3 According to the AFSPC *Enabling Concept for Space Situational Awareness*, achieving SSA presents many challenges including the following:

- Traditional joint ISR tasking, collection, processing, exploitation, and dissemination of space (threats) is fragmented among many separate national and DoD organizations and commands.
- Current intelligence assets and resources allocated to focus on persistent space system threats are insufficient to meet demands. SSA intelligence needs do not compete well within the National Intelligence Priority Framework.
- The existing SSN was not designed to meet, and is insufficient to support, space control needs (e.g., inadequate coverage to provide persistent surveillance of threats).

3.4 Operational Threat Environment. Possible threats to the overall Multi-TeleTrakNet system include ground attack, ballistic and cruise missiles, weapons of mass destruction, signals intelligence (interception of data), cyber attacks (unauthorized access by hackers or computer exploitation), electronic combat, espionage, terrorism, and sabotage to ground infrastructure. The Multi-TeleTrakNet systems' geographic locations within CONUS at Air Force installations place them in relatively non-hostile environment.

4. Synopsis.

4.1 **Missions.** The multi-TeleTrakNet system is a contributing sensor to the SSN having a primary mission of research, development and education supporting AFIT, and secondary mission of space surveillance supporting USSTRATCOM. The system will provide metric observations and SOI data. The system will supply data on cataloged objects, uncorrelated targets, and other targets as tasked by students and/or JSPOC. Space surveillance data will support JSPOC in its SSA responsibilities by providing tracking data to maintain the Space Order of Battle (SOB), Resident Space Object (RSO) catalog, as well as support Theater Operations. The multi-TeleTrakNet will transmit certified mission data to NASIC and JSPOC through JMS communications architecture. Multi-TeleTrakNet will expose data through the available net-centric capabilities.

4.1.2 Space Track. Multi-TeleTrakNet will provide observations to support completeness of the satellite catalog database. An accurate catalog is critical for maintaining the RSO catalog and the SOB, conjunction assessment, correct payload identification, treaty monitoring, decaying satellite notification, launch processing, on-orbit event detection and processing, monitoring space events and orbital debris analysis.

4.1.3 Space Intelligence. Multi-TeleTrakNet will provide metric and SOI data on LEO satellites to support DISOB, Battle Damage Assessment, Mission Payload Assessment, threat assessment, evaluation of satellite configuration and satellite anomaly resolution.

4.2 **Desired Effects.** Multi-TeleTrakNet will improve US space surveillance capabilities to find, track and characterize LEO objects in space from taskings from JSPOC. The system will improve SSN capabilities and support production of information usable throughout the full range of military operations for planning and execution. Multi-TeleTrakNet improves the following SSA capabilities:

- Tracking, identifying, and cataloging man-made objects in LEO orbit.
- Free other SSN sensor systems by accomplishing low priority taskings.
- Space Intelligence Preparation of the Operational Environment.
- High-fidelity SOI will contribute to the development and monitoring of space orders of battle—a critical component of space control operations planning and execution.
- Predictive Battlespace Awareness in space.
- Characterization of foreign space systems and space control forces will contribute to the development of a predictive, near real-time common operating picture of space.
- Notification of enemy military activities.
- Timely tracking and characterization of enemy space assets denies enemy forces the ability to covertly operate in space.
- Metric tracking data enables maneuver detection, conjunction assessment and targeting.
- SOI facilitates battle damage assessment.
- Real-time support to Offensive Space Control (OSC) or Defensive Space Control (DSC) operations.

4.3 **Operational View (OV).** Multi-TeleTrakNet will rate-tracking or sidereal tracking of LEO objects up to the six order in magnitude; SOI support to SSA; and tracking capability that can be tasked, as required, to support tactically responsive and flexible operations. OV-1 presented in thesis section 3.2.1.

5. Necessary Capabilities.

Timely and accurate detection, tracking, identification and characterization of man-made objects in space are necessary capabilities. In order to provide these capabilities the Multi-TeleTrakNet takes advantage of COTS equipment modified by AFIT faculty and students using the technology and equipment and exploits the inherent improvements to implement the following necessary capabilities:

5.1 Timely Detection of Space Events. Provide near real-time metric and SOI data for timely detection of space events such as OSC and DSC events (battle damage assessment), space launches, maneuvers, breakups, dockings, separations, reentries, decays, etc.

5.2 Accept and Respond to Special Tasking. Provide the ability to accept and respond to special tasking to support actions such as space event processing and OSC and DSC actions. The flexible and tactically responsive system will have the capability to respond as tasked to the situation. It is anticipated Multi-TeleTrakNet will follow current optical tasking codes.

5.3 Correlate Tracked Objects. Correlates tracked objects to the satellite catalog database. Provides the data to identify tracked objects as known or unknown to include objects in close proximity such as cluster tracking which reduces JSpOC cross-tagging and/or mis-tagging.

5.4 Provide Metric Data. Provide timely, accurate observations in the proper format to the C2 centers.

5.5 Interoperability with C2 Centers. Provide data interface(s) that allow interoperability with the C2 Centers.

5.6 Provide SOI Data. Provide photometric signatures (visual magnitude measurements over time) for SOI analysis. Provide visual magnitude data for use in support of optical sensor tasking.

6.0 Enabling Capabilities.

6.1 Communications Architecture. Multi-TeleTrakNet communications will flow through bases' communication nodes using dedicated, accredited, and encrypted, point-to-point circuits with C2 Centers.

6.2 Net-Centric Architecture. Multi-TeleTrakNet will expose data to the GIG. The GIG is the globally interconnected, end-to-end set of information capabilities for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel. The GIG includes owned and leased

communications and computing systems and services, software (including applications), data, security services, other associated services, and National Security Systems.

6.3 Training and Education. Air Education and Training Command, through the Air Force Institute of Technology Aerospace Engineering Department (ENY), will provide proficiency level system training to operating unit personnel (TBD after program expansion approval) and provide technical orders (operations and maintenance manuals) and training materials. Additionally, AFSPC, JSpOC and operating unit personnel will collectively build and implement TTPs that can utilize the capabilities of the multi-TeleTrakNet system to its fullest extent.

6.4 Security and Force Protection. The individual TeleTrakNet site locations will require a protection level assessment to determine the appropriate security and force protection requirements. This will ensure the proper resources are allocated to safeguard all TELETRAKNET assets to maintain the overall effectiveness of operations and make data available to operational customers.

6.5 Logistics. Multi-TeleTrakNet system will require a logistic support plan to provide integrated support of the system throughout its entire lifecycle. Multi-TeleTrakNet must have a capability to collect reliability and maintainability system information.

6.6 Manpower. Multi-TeleTrakNet manning will be contractor personnel. Manning requirements will be defined after program authorization.

6.7 Command and Control. Modernized C2 centers (JMS) will be necessary to ensure the AF can fully utilize multi-TeleTrakNet capabilities. Net-centric operations and open architectures must be utilized to support improved interoperability, collaborative sensor mission execution, non-stovepiped communications, efficient data transfer, and service-oriented operations.

7. Sequenced Actions.

7.1 General Description. Capabilities are required across the full range of military operations to provide space activity awareness that helps to protect the United States advantage in space and the ability to deny space capabilities to an adversary when directed. Multi-TeleTrakNet is specifically designed for optical rate-tracking and/or sidereal tracking during terminator conditions given a two- or three- line element set, and the object has a brightness magnitude lower than six and size larger than a basketball. The system provides metric and SOI data for directed tasking. The multi-TeleTrakNet system operates 24/7 with collection activation occurring thirty minutes prior to nautical twilight to thirty minutes after nautical sunrise. It will detect and/or track man-made objects within its field of regard during collection phase of operation. The multi-TeleTrakNet system will attempt to automatically correlate the objects against the JSpOC RSO catalog. Data that cannot be correlated to a known object will be disseminated in accordance with (IAW) JFCC SPACE (Unified Space Vault) direction and will be

processed IAW Strategic Command Directive 505-1 (Strategic Command Instruction (SI) 534-9 when published) and applicable directives. Multi-TeleTrakNet will also have the ability to perform hand-off operations to enable pass off of object from site to site. The system will respond to special tasking for long metric data tracks (i.e., temporarily focusing on one object instead of wide area survey) or SOI collection on high interest objects (e.g., unidentified objects, launches, OCS and DCS events, calibration satellites, etc.). Requests for taskings may originate from a number of sources such as combatant commanders or other government agencies. JSpOC will process these requests and task the multi-TeleTrakNet as needed in coordination with AETC research and development taskings. The system will support tasking from both machine-to-machine interface and manual input from an operator.

7.2 Day to Day Operations.

7.2.1 Multi-TeleTrakNet will operate in a remote mode with daily tasking schedule from AETC personnel and/or JSpOC, allowing the system to have a designated collection strategy to detect and track viewable objects. The data is correlated against the current catalog and transmitted to the C2 center via dedicated circuits and placed in a database by JSpOC to allow net-centric access IAW JFCC SPACE direction. Operational unit (TBD after program expansion) responsible to AFSPC will remotely monitor operations for those sites available for use (not being used by AETC personnel) and have the ability to interact with the sensor data. AETC personnel will identify site and telescope usage to operational unit and JSpOC the night prior to allow tasking optimization. If and while in sidereal tracking mode UCTs will be flagged and stored IAW established procedures. UCT disposition will be addressed after program expansion to assess the volume of UCT data the multi-TeleTrakNet system is capable of producing.

7.2.2 Since multi-TeleTrakNet will be remotely operated, immediate notification to operators of corrective maintenance situations is necessary to ensure the continuity of operations.

7.3 Communications and Data Integration.

7.3.1 The system will be managed in the SSN by the JSpOC, and exchange data in a secure and interoperable manner to enhance mission effectiveness. The multi-TeleTrakNet will interface with the C2 center via dedicated, accredited, and encrypted, point-to-point circuits including available net-centric capabilities. Through these networks it gains access to a complete and updated (daily) satellite catalog. This capability provides needed information for the system scheduler when the telescope is tasked.

7.3.2 It is envisioned that the multi-TeleTrakNet system will be fully integrated and able to be remotely operated from TBD AFSPC operational unit or remotely/locally operated from AETC personnel at a TeleTrakNet C2. In order to produce synergistic effects and maximize system performance, TTPs will be developed after program expansion as a

coordinated effort between, AFIT, HQ AFSPC, 614 Air and Space Operations Center (AOC), and the 21 SW. As a minimum, TTPs will address specific responsibilities for satellite hand-offs, TeleTrakNet site hand-offs, and UCT post processing. After program expansion approval, definitive relationships and operating concept for the multi-TeleTrakNet and AFSPC Operational Unit combined operations must be formulated.

7.4 Education and Training. To operate and fully exploit multi-TeleTrakNet capabilities requires trained operator and maintainer personnel. Training will cover all aspects of multi-TeleTrakNet operations. Site operators and maintainers will be provided initial training, and will conduct follow-on training IAW the contract statement of work. The unit shall be provided training materials, technical orders, commercial manuals, and/or operations manuals in sufficient time and in sufficient detail to support training. Technical orders and training material updates will be provided on an as needed basis to ensure training remains current with changes in equipment and operations.

7.5 Security and Force Protection. HQ AFSPC/A7/8 with the assistance of the 21 SW will develop the site/system protection level IAW Air Force Instruction (AFI) 31-101, *Air Force Installation Security Program*. Security for the TeleTrakNet sites is TBD but will be established and maintained IAW the applicable Department of Defense, Joint, Air Force, and AFSPC publications. Operations security, information security, and physical security policies and procedures will be integrated into appropriate unit-level procedures. HQ AFSPC/A3C will integrate the multi-TeleTrakNet system into the Contributing SSN Sensor Security Classification Guide (SCG). Program Protection Plans and/or SCGs must be understood and followed by all personnel with access to programs or program data.

7.6 Logistics.

7.6.1 AFIT will retain all logistics support material (e.g., System Engineering Plan, LCMP, Depot Source of Repair, etc.) required to plan, manage and provide lifecycle product support for multi-TeleTrakNet through memorandum of understanding agreements at participating bases. They will also ensure the multi-TeleTrakNet is maintainable and sustainable IAW AF policy.

7.6.2 Logistics support will use standard AF logistics structures and appropriate maintenance levels as defined in applicable AF 21- and 63- series instructions. The logistics concept will provide integrated support of systems throughout their entire life cycles. A two-level maintenance concept (organizational and depot) will be used. Acquisition and development of technical manuals for both operations and maintenance will be accomplished IAW Technical Order 00-5-3, *Lifecycle Management*.

7.6.3 Organizational or Level 1 maintenance of the TeleTrakNet sites will consist of removal and replacement of the Line Replaceable Unit(s), i.e., components which can be removed from the system without cutting or unsoldering connections. Additionally, repairs to external wires, cables, and connector repairs, and replacement of fuses, lamps, batteries, and other expendable items performed are considered organizational

maintenance tasks. Unless precluded by the operational situation, all authorized maintenance within the capability of a maintenance level will be accomplished before the equipment is sent to the next higher level.

7.6.4 Preventive Maintenance (PM) and Periodic Maintenance Inspections (PMI) will use developer provided technical orders (vendor/commercial off-the-shelf manuals) describing the frequency and procedures for accomplishing preventive maintenance and inspections, and replacement of time sensitive Line Replacement Units. PM/PMIs will be grouped into logical sequence and scheduled during non-mission time when possible to minimize impact on the operational mission. An effective PM program is essential to sustaining the operational system in a high level of availability. The PM program is designed to minimize outages and failures during the critical mission time. PM will be tracked and documented utilizing an integrated maintenance data system.

7.7 Manning. Multi-TeleTrakNet will be operated by O&M contractors under a TBD O&M contract or authorized AETC personnel. Specific manning requirements are not known at this time but will be based on mission requirements. Security forces manpower requirements, if any, will be determined as early as possible based on the protection level determination recommendation and the operating location.

8. Command Relationships. A memorandum of understanding upon program expansion approval would be developed between USSTRATCOM and AETC providing further details on the TTPs to be followed so the system can meet the separate goals of the two commands. The Unified Command Plan establishes USSTRATCOM as the functional combatant command for space operations. The Commander of USSTRATCOM (CDR USSTRATCOM) exercises Combatant Command of space forces as assigned in Section II of the Secretary of Defense (SECDEF) approved Global Force Management Implementation Guidance. In addition, CDR USSTRATCOM is responsible for providing military representation of space forces and space superiority to US national, commercial, and international agencies. This representation of forces applies to matters related to military space operations and as directed by the SECDEF and in coordination with the Chairman of the Joint Chiefs of Staff and appropriate combatant commanders.

8.1 Air Force Space Command (AFSPC). AFSPC is responsible to organize, equip and train USAF space forces required to achieve and sustain space superiority. AFSPC is the Component Major Command (MAJCOM) to USSTRATCOM for space; therefore, IAW USAF doctrine, the Commander, AFSPC is designated as the Commander, Air Force Forces (COMAFFOR) for space forces assigned to USSTRATCOM. AFSPC/CC presents space forces to CDR USSTRATCOM through the Commander of its component Numbered Air Force, 14 AF (AFSTRAT).

8.2 14 AF (AFSTRAT). 14 AF (AFSTRAT) is responsible for space operations (including supporting intelligence and communications) at the operational and tactical level. AFSPC/CC has delegated COMAFFOR day-to-day operational space responsibilities to the 14 AF (AFSTRAT)/CC. 14 AF (AFSTRAT) consists of both an

AOC and an AFFOR staff. Through its AOC, 14 AF (AFSTRAT) provides the capability to deliver operational space effects (global and theater-specific), products, and expertise to Combatant Commanders, Joint Force Commanders, service and functional component commanders, and other organizations supporting CDR USSTRATCOM and CDR JFCC SPACE.

8.3 Joint Functional Component Command for Space. CDR USSTRATCOM designated the Commander, 14 AF as CDR JFCC SPACE, and in his role is also USSTRATCOM's Space Coordinating Authority. CDR JFCC SPACE conducts space operations, exercises Operational Control (OPCON)/Tactical Control (TACON) of designated space and missile warning forces, and submits prioritized space operational requirements to CDR USSTRATCOM.

8.4 Joint Space Operations Center. The JSPOC serves as the 24-hour operations center for the execution of CDR JFCC SPACE responsibilities to plan, task, direct and monitor execution of joint space operations on behalf of CDR USSTRATCOM, Combatant Commanders, and supported and supporting organizations. JSPOC conducts 24/7 operations to support global space support, space control and space force enhancement missions. The SSA Operations Team and the ISR Operations Team will determine surveillance data needs for multi-TeleTrakNet throughout all test phases and operations, and coordinate tasking, surveys and data requirements. JSPOC, with coordination from NASIC and AETC personnel, tasks validated SOI requirements to the multi-TeleTrakNet system. JSPOC will generate metric observation and SOI tasking messages for multi-TeleTrakNet when system is not being used by AETC personnel for training, researching or educating purposes.

8.5 Air Education and Training Command (AETC). AETC's mission is to develop America's Airman through training institutions. Assists endeavor with memorandum of understanding between USSTRATCOM and AFSPC to provide access and support to a multi-TeleTrakNet system as a collateral sensor for the SSA.

8.6 Air University (AU). Headquartered at Maxwell AFB, Ala., conducts graduate education and professional continuing education for officers, enlisted members and civilians throughout their careers. Assists endeavor with memorandum of understanding between USSTRATCOM and AFSPC to provide access and support to a multi-TeleTrakNet system as a collateral sensor for the SSA.

8.7 Air Force Institute of Technology (AFIT). The current TeleTrakNet C2 and TeleTrak telescopes ground-based optical systems, located at the Wright-Patterson AFB, Ohio will enable new optical capabilities to support SSA. The TeleTrakNet capability will be added as a collateral sensor with responsibility of system when not in use to a TBD detachment within the 21 OG. Units will respond to taskings from JSPOC. The Air Force Institute of Technology primary mission to meet the ever changing and challenging scientific, engineering, and technical management needs of the Air Force and the Department of Defense through its graduate and continuing education programs.

8.8 United States Air Force Academy (USAFA). Potential future TeleTrakNet C2 and TeleTrak telescopes ground-based optical systems, located at the United States Air Force Academy, Colorado will enable new optical capabilities to support SSA. The TeleTrakNet capability will be added as a collateral sensor with responsibility of system when not in use to a TBD detachment within the 21 OG. Units will respond to taskings from JSpOC. The USAFA mission to expose cadets to a balanced curriculum that provides a general and professional foundation essential to a career Air Force officer takes precedence.

8.9 381st Training Group. Potential future TeleTrakNet C2 and TeleTrak telescopes ground-based optical systems, located at the Vandenberg AFB, California will enable new optical capabilities to support SSA. The TeleTrakNet capability will be added as a collateral sensor with responsibility of system when not in use to a TBD detachment within the 21 OG. Units will respond to taskings from JSpOC. The 381st Training Group's mission to prepare space and missile operators through a specific portion of formal technical training required to accomplish the Air Force mission will be enhanced through such an optical telescope system.

8.10 National Air and Space Intelligence Center (NASIC). In support of USSTRATCOM and JFCC SPACE, NASIC's responsibilities include assessing and characterizing foreign space systems capabilities, processing, exploitation and dissemination of SOI data, and maintaining the Space Order of Battle.

9. Summary. AFIT is developing and demonstrating a capability that shows great promise to space control operations. A multi-TeleTrakNet system could provide operationally relevant surveillance and intelligence information on friendly and adversary space systems to produce actionable information that can be used at all levels of warfare for both planning and execution. These improvements in surveillance capability directly tie to improvements in SSA and supporting OSC, DSC, and ultimately space superiority. A multi-TeleTrakNet will:

- Provide visual magnitude data
- Provide photometric signatures
- Provide optical SOI data
- Provide remote operations
- Provide sensor hand-off operations
- Support net-centric capability
- Support improved timeliness of space event processing
- Provide high accuracy metric observations and support orbit determination
- Support improved RSO catalog completeness by allowing other systems to focus on tracking smaller objects
- Support space superiority, resulting in offensive and defensive advantages
- Support predictive battlespace awareness

10. CONOPS Appendix.

Strategic Command Directive 505-1, *Space Surveillance Operations* 13 Feb 04

SI 508-10, *Mission Integrity, Change Control Management, and Test*

Control for the ITW/AA Systems 12 Jul 06

USSTRATCOM *Joint Capabilities Document for Space Control* 22 Jul 06

Air Force *Space & C4ISR Concept of Operations* 28 Apr 06

Air Force Technical Order 00-5-3, *Lifecycle Management* 1 Apr 98

AFI 31-101, *The Air Force Installation Security Program* 1 Mar 03

AFSPC *Enabling Concept for Space Situational Awareness* 22 Oct 07

AFSPC *Functional Concept for Space Control Operations* 12 Feb 08

AFSPCI 10-604, *Space Operations Weapon System Management* 1 Oct 07

Dedicated SSN Sensors SCG 30 Sep 05

Appendix B. TT2k15_serial_tester_v3

```
%script to test new manner of serial handling in R2014...

close all
clear all
clc

%start by establishing scope info - this appears to work fine
[scopepresent,scopecom] = findtelescope;

fclose(scopecom);

set(scopecom,...
    'ReadAsyncMode','continuous',...
    'BytesAvailableFcnMode','byte',... %was 'terminator'
    'BytesAvailableFcnCount',10,... %hard-coded to az value
    'BytesAvailableFcn',[...
        'toc',...
        ',',... disp('For the read:'),',...
        ',',... 'tic',',...
        '[scopestring,count,msg] = fscanf(scopecom,'%c',10);',... %[scopestring,count,msg] = fgets(scopecom);',...
        ',',... 'toc',',...
        'time(rx,3) = length(scopestring);',...
        'time(rx,4) = meadestring_to_angle(scopestring);',...
        'ct_rx = clock;',...
        'time(rx,2) = ct_rx(5)*60+ct_rx(6);',...
        'rx = rx + 1;',...
    ''])

scopecom.RecordDetail = 'verbose';
scopecom.RecordMode = 'index';
scopecom.RecordName = 'serialstuff.txt';

fopen(scopecom)

record(scopecom)

period = 0.5; %some values are terrible, no clear pattern ID'd
    %this is MATLAB / serial, nothing to do with the scope
    %restarting MATLAB cleared it (1x)
    %failing to close the serial object does not cause it (1x)
    %when bad,
    %for FixedRate
    %0.035, 0.1 are great
```

```

    %0.15, 0.2, 0.25, 0.35, 0.45 are terrible
    %0.5 is iffy,
    %0.55 worked well at least once instance
    %0.7 ratty, +/- 0.4sec
    %0.75 is good, +/- 0.15 sec, breaks down after 30 trials (?)
    %1 seems bad, +/- 0.3 sec
    %2 is great; +/- 150 msec or so, scope seems sleepy
%for FixedSpacing
    %0.1 goes unstable / bursty (0.5 to 1 sec lags)
    %0.3 runs smooth, but about 0.55 sec b/w txmits
    %0.55 lags about 0.1sec (0.65 b/w txmits)
%for FixedDelay
    %0.1 goes unstable / bursty (0.5 sec lags)
    %0.3 is terrible
    %0.5 is pretty good

```

```

azturn = true; %flips between az & el each timer execution
tasks = 500;
time = zeros(tasks,4);
tx = 1;
rx = 1;
azrate_send = 0;
slewdirection_az = 'e'; % Change slew direction east instead so it can
%start at 0 and increase positive numbers

```

```

systemtimer = timer('Period',period,...
    'Name','systemtimer',...
    'ExecutionMode','FixedRate',...
    'TaskstoExecute',tasks,...
    'TimerFcn',[...
        'if tx <= 3 || tx == tasks;',...
        'azrate_send = 0;',...
        'else;',...
        'azrate_send = tx*0.01;',...
        'end;',...
        'fprintf(scopecom,['':RA"',sprintf('"%0.2f"',azrate_send),'#':M"',slewdirection_az,'#':GZ#'''],',...
        'tic;',...
        'ct_tx = clock;',...
        'time(tx,1:3) = ct_tx(5)*60+ct_tx(6);',...
        'tx = tx + 1;',...
    ],...
    'StopFcn',[]);

start(systemtimer)

```

[Published with MATLAB® R2013a](#)

Appendix C. TT2k15_visible_clock

```
%This function create a "clock" suitable for filming with a webcam while
%using a _trackgui variant. It can be run in parallel with said trackgui
%on a separate monitor.

%In early 2015, Meade telescope calibration scripting evolved to the point
%where it was unclear what delays, if any, existed between the exposure
>window defined by MATLAB (i.e., event.Timestamp in IMAQ's built-in
>acquisition tool). Earlier work only confirmed that event.Timestamp was
>the only "useful" time data available to each video frame, since it could
>show a timestamp quite disparate from a call to the clock() function at
>the same time.

%To estimate total delay in a video stream, a sawtooth pattern with
>endpoints corresponding to whole seconds at the left and right screen
>edges, respectively, is displayed onto a monitor. On reconstruction of
>the video stream, a sawtooth of known period, in this case one second, can
>be curve-fit to the recorded data. The offset or delay is determined by
>comparing the fit peaks to their occurrence after a peak integer value.

%There are some assumptions in this approach:
% -Delays will be less than one second (results become ambiguous).
% -There is a limit on the approach due to:
%   -Monitor refresh rate (e.g, 60Hz = 16 msec resolution).
%   -MATLAB's single-threaded nature may delay timer processing;
%   working precision may be something like 10msec, which is sometimes
%   cited as the "Windows limit" for a single process thread.
% -It might be worse when running the GUI; other approaches could
%   include starting a second MATLAB instance or writing a similar
%   piece of code in a separate programming language on a separate
%   thread. Simulink supposedly has better "realtime" capability as
%   well. Since this is an initial look, these other options are
%   noted for completeness only.

% Matt Schmunk, Jan 15
% clear all; close all; clc
visclockfps = 20;

%MATLAB's min res is 1msec, this appears to be the working precision of
>clock() under most circumstances as well (although I've seen it at 1e-6
>sec res using direct calls from the command line). No matter, msec is
>more than sufficient for this level of test.
```

```

%set up to show the target for only half a frame; reduces probability of
%a streaking target...
visclockperiod = roundn(1/(2*visclockfps),-3); %in sec

visclockflipper = 0;

visclocktimer = timer(...
    'BusyMode','drop',...
    'ExecutionMode','fixedRate',...
    'Name','visible_clock_timer',...
    'Period',visclockperiod,...
    'Tag','visible_clock_timer',...
    'TimerFcn',[...
        'if visclockflipper == 0,... %off cycle
            'set(visclocktarget,'MarkerFaceColor','k','MarkerEdgeColor','k'),...
            'visclockflipper = 1;',...
        'else,... %on cycle
            'visclock = clock;',...
            'set(visclocktarget,'XData',0.49 + (1-2*0.49)*mod(visclock(1,6),1)),',...
            'set(visclocktarget,'MarkerFaceColor','w','MarkerEdgeColor','w'),',...
            'visclockflipper = 0;',...
        'end',...
        ',... set(vidclockaxes,"Position",',...
        ',...'[0.35 + (1-2*0.35)*mod(visclock(1,6),1)-0.005,0.495,0.01,0.01]),',...
        ',... fprintf("Fraction of a sec is %1.3f.\n",mod(visclock(1,6),1)),',...
    ]...
);

visclockfigure = figure(...
    'Color','k',...
    'MenuBar','none',...
    'NumberTitle','off',...
    'Name','Visible Clock',...
    'Position',[0 900 1680 1050],...
    'ToolBar','none',...
    'Tag','visclockfigure_1',...
    'Units','Pixels',...
    'CloseRequestFcn',[...
        'if invalid(visclocktimer) == 1,...
            'stop(visclocktimer),',...
            'delete(visclocktimer),',...
            'clear visclocktimer visclockflipper,',...
        'end',...
        'clear visclock visclockfps visclockperiod,',...
        'delete(visclockfigure),',...
    ]...
);

```

```

    'clear visclockfigure visclockaxes visclocktarget',...
    ]);

    %this is the axes
    visclockaxes = axes('Parent',visclockfigure,'Units','Normalized',...
        'Position',[0 0 1 1],'Color','k');
    set(visclockaxes,'XTick',[],'YTick',[],'XTickLabel',[],'YTickLabel',[]);
    set(visclockaxes,'YLim',[0 1],'XLim',[0 1],'DrawMode','fast')

    %this is the target
    set(0,'CurrentFigure',visclockfigure);
    set(visclockfigure,'CurrentAxes',visclockaxes); %allows refresh without
        %changing figure state (means figure doesn't get focus on refresh)
    hold on
    visclocktarget = plot(0.5,0.5,'ws','MarkerFaceColor','w','Parent',visclockaxes,'MarkerSize',1);
    hold off

    start(visclocktimer)

    return

```

Appendix D. How to create a precalcs.mat file

Acquire a set of two line e from www.space-track.org

It requires having a username and passcode

Once logged on click on the [Recent TLEs](#) tab to choose a desired catalog



The Recent TLEs tab has all if not most the current satellites neatly packed in several categories. This is mission driven because downloading the whole catalog will take more space although the precalcs.m file can perform this very quickly.

The precalcs.m file uses the Three Line Element

When choosing a catalog a new window will be displayed.

Wait for the whole list to be displayed on the page.

Copy and paste all of the data onto a notepad and save it on the TLE folder for ease.

The current nomenclature for saving the file has been YYYYMMDD_3le.txt

Running the TT2k15_precacals.m file

The following are the current questions that the precalcs.m script requires to correctly determine the location of the telescope in relation to the expected location of the satellites.

Choose a location

Currently there are four options: Big Guns, Rooftop, Matt and Other

The first three are static locations but it has an option to manually input the current location of the telescope.

Define the day of the observation

Handy if you would like to estimate “tomorrow’s” paths it can be done by adding a day or subtract a day if you forgot what could have been seen at a specific time.

Choose the correct file

It is possible to choose a different day since precalcs.m doesn’t know the difference.

Determine the maximum satellite period.

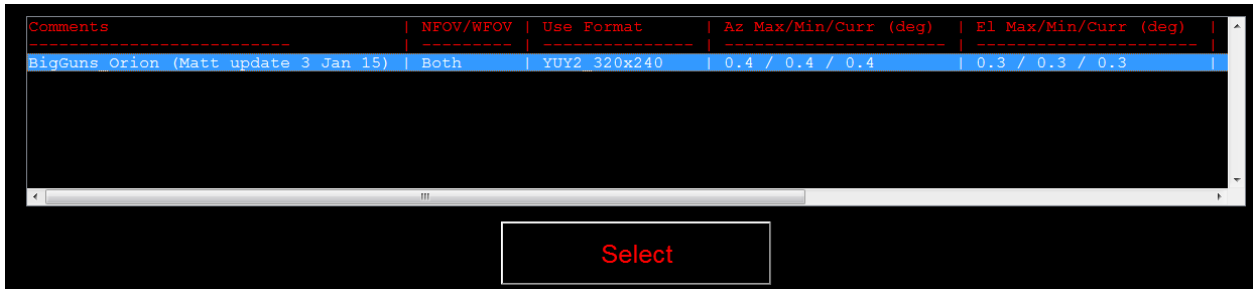
Determine the brightness of the objects of interest

Determine the elevation threshold

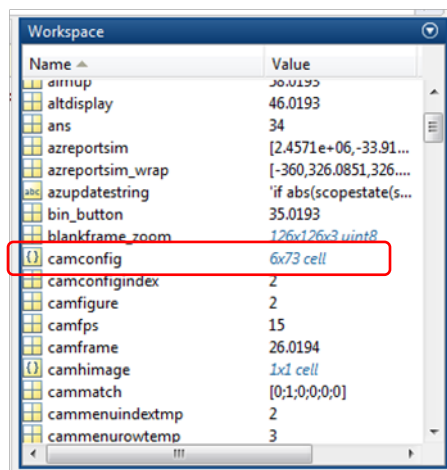
Once the required information is set, the precalcs script will create the precalc.mat file necessary if the object to be observed is known.

Appendix E. How to configure the digital camera

- Run the TT2k15_trackgui and choose your camera or if is a new camera, choose the default camera and click “Select” to continue



- Once the GUI is running look at the workspace on the main MATLAB program (not the GUI) and double click the camconfig structure and a new spreadsheet will open



- In this spreadsheet you will need to ensure the following settings
 - Camera Name
 - Comments – Provide a meaningful way to differentiate this camera settings for the specific location and optics
 - Use Format – to change this double click on the “Supported Formats” cell next to open a new sheet and copy the desired format

- Az FOV Max, Az FOV Min, Az FOV Current as well as its counter parts for El FOV – If not known, this value can be calculated via the GUI by setting a static target in the center of the screen. From this point, move the telescope very slowly or move the target until is almost out of the screen's view. The FOV is the distance a target travels from the center of the screen to the end times 2. This distance is noticed in the +R and +U values. Example if the target at the edge of the screen is +R: .2 and +U: .15 then the FOV Max is 0.4 for Az and 0.3 for El

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	'Camera Name'	'Comments'	'Supported ...'	'Use Format'	'Az FOV Ma...	'Az FOV Mi...	'Az FOV Cur...	'El FOV Max...	'El FOV Min...	'El FOV Curr...	'Camera Cal'	'Is NFOV?'	'Is WFOV?'	'Flip?'	'B
2	'Philips SPC 900NC PC Camera'	'BigGuns_O...	1x15 cell	'YUY2_320x...	0.4100	0.4100	0.4100	0.3000	0.3000	0.3000	[]	1	1	0	[0,
3	'Logitech QuickCam Pro 4000'	'BigGuns_M...	1x15 cell	'RGB24_640...	40	30	40	30	22.5000	30	[]	0	1	0	[0,
4	'USB2.0 2MP UVC AF Camera'	'Matt's PV...	1x15 cell	'YUY2_320x...	4.6000	4.6000	4.6000	3.2000	3.2000	3.2000	[]	1	0	0	[0,
5	'Built-in iSight'	'Matt's Ma...	1x5 cell	'YUY2_640x...	40	30	40	30	22.5000	30	[]	0	1	0	[0,
6	'USB2.0 HD UVC Camera'	'Matt's pro...	1x14 cell	'YUY2_640x...	46	46	46	34.5000	34.5000	34.5000	[]	0	1	0	[0,
7															
8															

- NFOV – if the camera is to be used for narrow field of view enter 1 for yes (0 for no)
- WFOV – 1 always
- To set the correct frame of the camera first must determine which cell is responsible for this – look at “Device Field” cell 27 and double click. An example is below.

camconfig

6x73 cell

27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
'Device Fiel...	'Dev. Field 1'	'Use Dev. Fi...	'Dev. Field 2'	'Use Dev. Fi...	'Dev. Field 3'	'Use Dev. Fi...	'Dev. Field 4'	'Use Dev. Fi...	'Dev. Field 5'	'Use Dev. Fi...	'Dev. Field 6'	'Use Dev. Fi...	'Dev. Field 7'	'Use Dev. Fi...
13x1 cell	1x1 struct	'on'	1x1 struct	36	1x1 struct	'on'	1x1 struct	32	1x1 struct	'15.0000'	1x1 struct	'15.0000'	1x1 struct	[]
23x1 cell	1x1 struct	'on'	1x1 struct	750	1x1 struct	'on'	1x1 struct	5056	1x1 struct	3	1x1 struct	'15.0000'	1x1 struct	'30.0000'
23x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	20	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	40	1x1 struct	'auto'
17x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	32	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	'15.0000'	1x1 struct	220
18x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	15	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	'10.0000'	1x1 struct	0

camconfig

camconfig(2, 27)

13x1 cell

	1	2	3
1	BacklightC...		
2	Brightness		
3	ColorEnable		
4	Contrast		
5	FrameRate		
6	Gamma		
7	Parent		
8	Saturation		
9	Selected		
10	SourceName		
11	Tag		
12	Type		
13	VerticalFlip		

- In this example the Frame Rate is in cell 5
- For this example double click on the camconfig spreadsheet on “Dev Field 5” to determine the many frame options under the “ConstraintValue” cell

camconfig

camconfig(2, 27)

camconfig(2, 36)

1x1 struct with 6 fields

Field	Value
Type	'string'
Constraint	'enum'
ConstraintValue	1x7 cell
DefaultValue	'50.0000'
ReadOnly	'whileRunning'
DeviceSpecific	1

- Highlight the desired frame rate and copy it

camconfig							
camconfig{2, 27}							
camconfig{2, 36}							
camconfig{2, 36}.ConstraintVal							
1x7 cell	1	2	3	4	5	6	7
1	60.0002	30.0000	25.0000	20.0000	15.0000	10.0000	5.0000
2							
3							
4							

- o Go back to the camconfig spreadsheet and choose the cell to the right of the “Dev Field #” where the Frame Rate was found – in this example it is the “Use Dev Field 5” and paste the value. Note: you won’t be able to just type the desired value

camconfig														
6x73 cell														
27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
'Device Fiel...	'Dev. Field 1'	'Use Dev. Fi...	'Dev. Field 2'	'Use Dev. Fi...	'Dev. Field 3'	'Use Dev. Fi...	'Dev. Field 4'	'Use Dev. Fi...	'Dev. Field 5'	'Use Dev. Fi...	'Dev. Field 6'	'Use Dev. Fi...	'Dev. Field 7'	'Use Dev. Fi...
13x1 cell	1x1 struct	'on'	1x1 struct	36	1x1 struct	'on'	1x1 struct	32	1x1 struct	'15.0000'	1x1 struct	'15.0000'	1x1 struct	[]
23x1 cell	1x1 struct	'on'	1x1 struct	750	1x1 struct	'on'	1x1 struct	5056	1x1 struct	'5'	1x1 struct	'15.0000'	1x1 struct	'30.0000'
23x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	20	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	40	1x1 struct	'auto'
17x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	32	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	'15.0000'	1x1 struct	220
18x1 cell	1x1 struct	'on'	1x1 struct	0	1x1 struct	15	1x1 struct	-5	1x1 struct	'auto'	1x1 struct	'10.0000'	1x1 struct	0

- The last step on the digital camera configuration is to save the file using the following code on the Command Window of MATLAB – save camconfig.mat camconfig.

Appendix F. A day in the life of iTeleTrak “to-be”

Pre-processing

- A request for the collection of a resident space object (RSO)
- A team member:
 - a) Download the most current TLE from CelesTrak, <http://celestrak.com/index.asp> or Space-Track, www.spacetrack.org (Space-Track is currently the preferred method). If an element set from another source is available (e.g., derived at AFIT from previous observations), that element set can also be processed subject to the assumptions below.
 - b) Uses precalcs.m to generate tracking data for acceptable RSO(s) for the specific research goal, noting the following:
 - By default, only RSOs with public-domain brightness data are analyzed (see getcatalogsats.m for details); brightness database files must be periodically updated by the team member.
 - Objects without public-domain data can be added to specials.txt, a database file that overrides the default assumption (for example, if attempting to observe a recently launched object).
 - c) Reviews/rehearses precalculated track data; the trackgui program allows the team member to run at a “simulated” time, to check the results and/or practice what might be experienced during an observation session.

Pre-collection phase

- Starts 1 – ½ hour prior to desired RSO track to get the equipment ready
- Terminator conditions are achieved when the telescope is in darkness and the RSO is still in sunlight. Because of this, it is best to do collections right after sunset or before sunrise (with a 2 hr window)
- The team member accomplishes set up, bore-sight alignment, and system checkout for rooftop and/or machine shop.
- Team member in the ASOC
 - Remotes into each site and tests the remote capability while in contact with the team member at the telescope
 - Initiates commands to the selected telescope to slew to the projected target’s azimuth and elevation position and it is repeated for any additional telescopes
- Each telescope uses the calculated AZ-El data based from the TLE to begin preliminary rate tracking of RSO target. Adjustments to the track can be performed in a few different ways (using at a minimum a guide scope only, or also the main optic’s camera if connected to the trackgui):
 - Manual “offset” panning of the telescope, i.e. track so many degrees above, below, left, or right of the calculated az-el data

- Manual “in-track” adjustments to the telescope; corrects for an “early” or “late” arriving spacecraft and/or time calibration issues on the telescope
 - Semi-automated video tracking: “track where I click on the video.”
 - Fully-automated video tracking “when locked on a bright object, move it to the screen center”
- In this mode the stars streak across in the background.
- The team member at the telescope communicates with the ASOC to achieve any necessary adjustments to get the RSO target into the spotting telescope FOV using the Meade viewfinder and cell phone or radio
- If the main optic is not connected to the trackgui, the ASOC can inform the remote operator when the target is within the main optic’s estimated field of view, which is displayed as a smaller box on the trackgui video screen.

Collection phase

- The ASOC team member then selects the record data button to begin video capture of whichever camera is selected in the trackgui.
 - If two cameras are present, and the “tracking” camera is switched, the trackgui will record whichever camera is selected.
 - If an “offline” system is used to collect data (i.e. from the main optic), the remote operator would have to start recording using that system.
- The same process is performed for the additional telescopes
- The team member at the telescope site adjusts the main optic focus as needed according to direction from ASOC. (Note a RoboFocus automated focuser had been previously integrated into TeleTrack, but is not currently available pending further shakedown).
- ASOC adjusts the gain and telescope AZ/EL position if needed if closed-loop tracking is lost during the length of time the RSO is in view through the optics
- ASOC and or team member will determine when to stop recording
- ASOC then selects a new RSO to be tracked according to the plan or as a target of opportunity
- Once all the RSOs in the plan are attempted, the end of terminator is reached, or the collection goals are met the site is then returned to safe mode by:
 - Placing the telescopes to the home position
 - Placing lens caps on the scopes
 - Lowering the telescope pier
 - Closing roof at each site

Post-processing

- There can be a few approaches to post-processing, depending on the research objective.
 - If only the telescope’s tracking data is of interest, a .csv file containing telescope parameters can be analyzed in MATLAB or Excel. This file contains a variety of timestamps, telescope positions, and tracking states. For example, in orbit

determination applications a student might only be interested in track data collected when the target was “locked” and within a pre-determined range of degrees of boresight. The “playback” code makes it easier to delog this data.

- If video data collected in the trackgui is of primary interest, the “playback” code can be used to decompress it, analyze it, convert it to another format (e.g., AVI), etc. The “playback” code also allows cross-correlation to other logged data as noted above.
- If another offline tool was used to collect data, then that system’s post-processing techniques are applied. It should be assumed that this kind of collect cannot be as precisely time-correlated to the other two sources noted above.
- Data should be saved to the central TeleTrak hard drive, noting the following:
 - Data collected using the trackgui is relatively “self-filing,” that is the time and object tracked are automatically placed in the filename(s) for convenience.
 - Data collected using offline systems should be renamed using a similar convention, using the start time of the recording and the object number tracked at a minimum.

Appendix G. Explanation of the sawtooth test

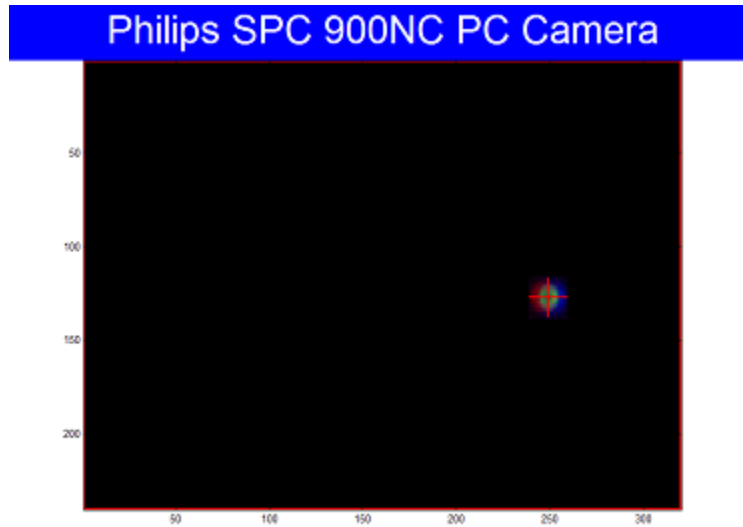


Figure 1

Figure 1 shows one frame of the filmed `visible_clock.m` output; the crosshair marks the center of the square-pixel target that sweeps across the screen once per second, left-to-right. If more than one dot appears in the frame due to the length of the exposure, then the rightmost object is selected.

Next, the `playback.m` code, set to Camera Cal mode, assumes that the camera filmed a true sawtooth. The true sawtooth is composed so it has a period of one second, and an amplitude matching the size of the filmed sweep. The amplitude is calculated by subtracting the maximum value of all detected dots from the minimum of all detected dots.

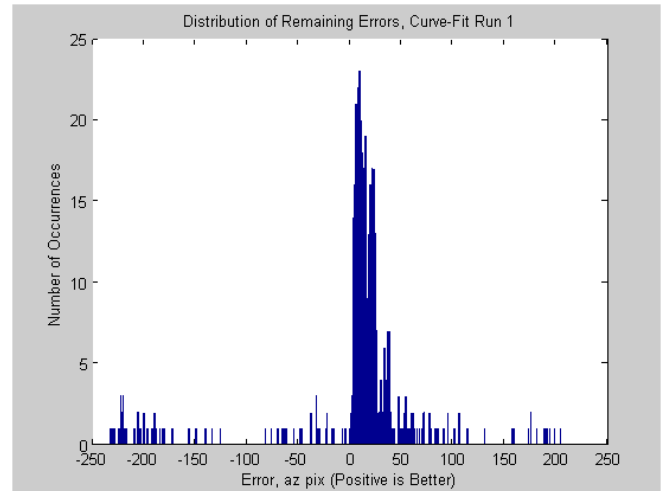
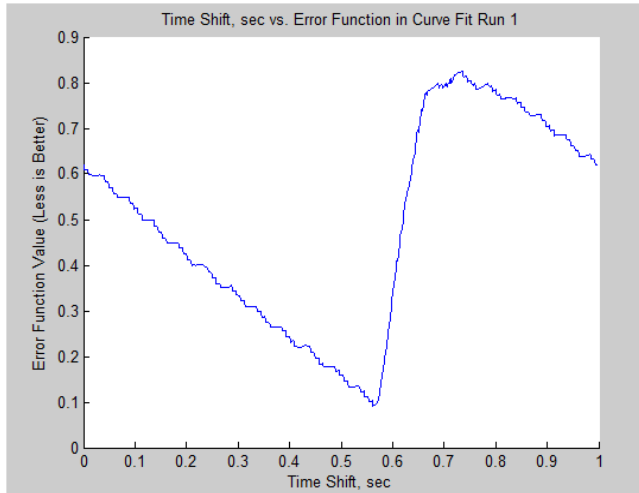
To align the true sawtooth, a two-step process tries to minimize the distance between the detected dots and the true sawtooth. If the curve is “ahead” of the dot this is good, with an error function defined as:

$$\text{residual} = \text{curve position} - \text{detected dot (+ right)}$$

A positive value is good, because it's possible that the dot had not quite yet appeared when the frame was exposed. Negative is bad, because it means the dot appeared before a one-second sweep would have. The code makes an early attempt at the optimal fit by counting the number of negative errors that occurred at a time shift (`t_shift`) between 0 and 1000 milliseconds. The results are stored as a percentage, brute-force style, and after all the runs are complete the minimum value (the minimum number of negative errors occurred). The equation that tallies each percentage is:

negative error = sum(residuals < 0)/size(residuals,1);

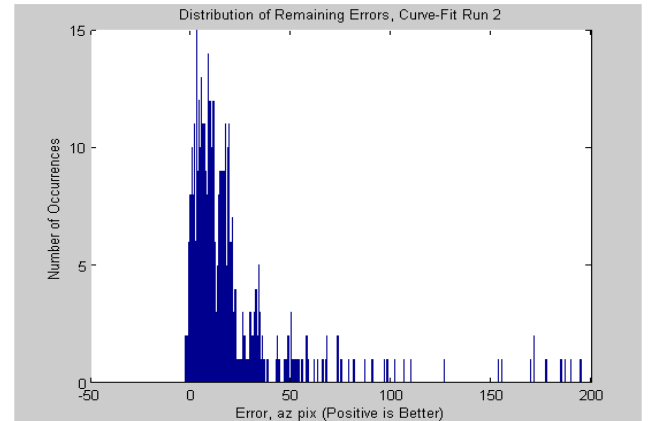
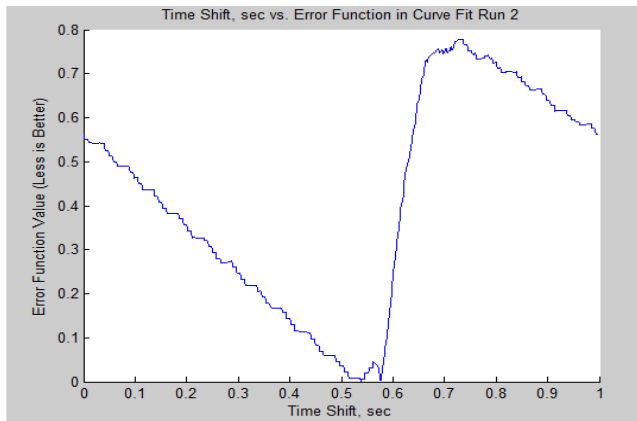
After the min is selected, a histogram displayed to help confirm that most results are, in fact, negative (in this example, the minimum occurs at about 0.57 seconds from the arbitrary start of the recording).



Extreme outliers are removed from the data set; only positive errors are retained and re-evaluated in the second run. Run two repeats the same basic brute-force process, except that instead of simply counting the number of negative errors, the goal is to ensure that some statistical “cutoff” (2-sigma, ≈ 0.9545) occurs. The equation that tallies each percentage on this run is:

negative error = abs((1-cutoff) - sum(residuals < 0)/size(residuals,1))

The consequence of this approach is that there can be two minimums, but using MATLAB's “find” function the algorithm always selects the right-most side, which ensures that the number of detects occurring *after* the sawtooth fit is met. Again, a histogram shows the result to confirm success:



The second run changes the initial results only slightly. Now that a curve fit is established, the system delay can be easily computed by comparing where the peak value of the sawtooth falls with respect to the nearest integer second. Since we know the right-most dot appeared at the end of a whole integer second, and the peak value of the sawtooth represents when the camera *recorded* it arriving, then the time delay of the camera is essentially calculated like this:

$$\text{camera delay} = \text{time of sawtooth peak} - \text{floor}(\text{time of sawtooth peak})$$

[floor is the same as modulo 0, or rounding down to the nearest integer]

Bibliography

- [1] US Air Force, Air Force Doctrine Document 3-14, Space Operations, LeMay Center for Doctrine Development and Education, 2012.
- [2] U.S. Joint Chiefs of Staff, "Space Operations, Joint Publication 3-14," U.S. Joint Chiefs of Staff, Washington, DC, 2013.
- [3] M. W. Schreiner, "Discrete event simulation of the low earth orbit satellite tracking telescope network: TeleTrakNet," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2012.
- [4] M. M. Schmunk, "Initial Determination of Low Earth Orbits Using Commercial Telescopes," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2008.
- [5] U.S. Department of Defense, "The National Security Space Strategy," U.S. Department of Defense, Washington, DC, 2011.
- [6] USSTRATCOM, "USSTRATCOM Space Control and Space Surveillance," [Online]. Available: http://www.stratcom.mil/factsheets/11/Space_Control_and_Space_Surveillance/. [Accessed 22 February 2015].
- [7] B. M. Bennett, "Systems Engineering Approach to Automated Cueing for LEO Satellite Tracking," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2014.
- [8] National Aeronautics and Space Administration (NASA), "Definition of Two-line Element Set Coordinate System," 23 September 2011. [Online]. Available: http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html. [Accessed 1 March 2015].
- [9] US Air Force, "US Air Force Fact Sheet: Ground-Based Electro-Optical Deep Space Surveillance," US Air Force, Peterson AFB, 2008.

- [10] G. C. Briggs, "Satellite Detection and Real-time Orbit Estimation With Commercial Telescopes," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2011.
- [11] P. F. Sydney, J. L. Africano, A. Fredericks, K. M. Hamada and V. S. Hoo, "Raven automated small telescope systems," in *Imaging Technology and Telescopes*, San Diego, CA, 2000.
- [12] Public Groups, "LX200 GPS," Yahoo Groups, 10 June 2013. [Online]. Available: <https://groups.yahoo.com/neo/groups/lx200gps/conversations/messages/128217>. [Accessed 9 March 2015].
- [13] Meade Instruments Corporation, Instruction Manual: 8", 10", 12", 14", 16" LX200GPS Schmidt-Cassegrain Telescopes with AutostarII Hand Controller, Irvine: Meade Instruments Corporation, 2003.
- [14] P. Murphy, "Laser effects on cameras and camcorders," LaserPointerSafety.com, 22 October 2014. [Online]. Available: <http://www.laserpointersafety.com/styled-6/ilda.html>. [Accessed 10 March 2015].
- [15] J. L. Gresham, "Optical Closed-Loop Control Tracking with Commercial Telescopes," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2012.
- [16] M. E. Graff, "Development of a remotely operated autonomous satellite tracking system," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2010.
- [17] J. W. Driskell, "Low Earth Orbit Satellite Tracking Telescope Network: TeleTrakNet," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2011.
- [18] C. L. Carlton, "Lucky Imaging of Low Earth Orbit Satellites," MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, 2009.

- [19] INCOSE, "What is Systems Engineering?," 14 June 2004. [Online]. Available: <http://www.incose.org/practice/whatisystemseng.aspx>. [Accessed 1 March 2015].
- [20] B. S. Blanchard & W. J. Fabrycky, Systems Engineering and Analysis Fifth Edition, Upper Saddle River: Prentice Hall International, 2011.
- [21] Software Testing Class, "V Model," 29 May 2012. [Online]. Available: <http://www.softwaretestingclass.com/v-model>. [Accessed 1 March 2015].
- [22] C. Larman, An Introduction to Object-Oriented Analysis and Design and Iterative Development, Upper Saddle River: Prentice Hall PTR, 2005.
- [23] Department of Defense Chief Information Officer, "Department of Defense Architecture Framework (DoDAF), Version 2.0," US Department of Defense, 2009.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 27-03-2015		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Oct 2014 - March 2015	
TITLE AND SUBTITLE Low Earth Orbit Satellite Tracking Telescope Network: Collaborative Optical Tracking for Enhanced Space Situational Awareness				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Salvador, Victor A., TSgt, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENV-MS-15-M-200	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RV 3550 Aberdeen Ave SE Kirtland AFB, NM 87117				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RV	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT The Air Force Institute of Technology has spent the last seven years conducting research on orbit identification and object characterization of space objects through the use of commercial-off-the-shelf hardware systems controlled via custom software routines, referred to simply as TeleTrak. Year after year, depending on the research objectives, students have added or modified the system's hardware and software to achieve their individual research objectives. In the last year, due to operating system and software upgrades, TeleTrak became inoperable. Furthermore, due to a lack of student overlap, knowledge of the basic operation of the TeleTrak deteriorated. This research re-establishes the basic understanding of the TeleTrak System and develops a plan to improve the telescope tracking controller performance. This research uses a subset of the SE process via the operational and system views to understand the tracking subsystem and develop timing tests to observe delays that could impact tracking. Basic tests revalidate and improve understanding of how the Meade telescopes interface with MATLAB. Calibration camera parameters are then refined, allowing a new technique for calibrating existing control algorithms. The analyses of the findings demonstrate that it is possible to improve the tracking controller, but it also uncovers previously undocumented issues with the Meade telescope mount. Future students interested in continuing this research, regardless of which telescope mount is used with TeleTrak, will benefit from the findings of this research.					
15. SUBJECT TERMS COTS Telescope, Ground-Based Satellite Imaging, TeleTrak, TeleTrakNet, Satellite Tracking, Space Situational Awareness (SSA), Space Surveillance Network (SSN), Two Line Element Set (TLE)					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 108	19a. NAME OF RESPONSIBLE PERSON Dr. Richard G. Cobb, AFIT/ENY
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4559 (richard.cobb@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18